HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology

Nassir Workicho Oumer

Development of Wireless Control System for a Spherical Robot

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Espoo August 19, 2009

Supervisors:

Professor Aarne Halme
Helsinki University of Technology

Professor Kalevi Hyyppä
Luleå University of Technology

Instructor:

Tomi Ylikorpi
Helsinki University of Technology

Acknowledgements

I would like to acknowledge Professor Aarne Halme, of the Automation and Systems Technology Department, TKK, Finland, for providing this great opportunity to research and experiment in the interesting field of spherical robots as well as for his important comments on my work. Also, I wish to acknowledge Professor Kalevi Hyyppä, Department of Computer Science and Electrical Engineering, Luleå University of Technology, Sweden, for his invaluable comments at every milestone of the research.

I would like to express profound gratitude particularly to Mr. Tomi Ylikorpi for his invaluable support, encouragement, supervision and useful suggestions throughout this research work as well as coordinating SpaceMaster studies at TKK.

Also, I gratefully appreciate the language support offered by Mr. William Martin, Department of Networking and Communications, TKK towards the completion of the thesis. The improved readability of the thesis would not be as it is now without his support.

Espoo, August 19, 2009

Nassir Workicho Oumer

Helsinki University of Technology Abstract of the Master's Thesis

Author:	Nassir Workicho Oumer
Title of the thesis:	Development of Wireless Control System for a Spherical Robot
Date:	August 19, 2009 **Number of pages:** 83
Faculty:	Faculty of Electronics, Communications and Automation
Department:	Automation and System Technology
Program:	Master's Degree Programme in Space Science and Technology
Professorship:	Automation Technology (Aut-84)
Supervisors:	Professor Aarne Halme (TKK)
	Professor Kalevi Hyyppä (LTU)
Instructor:	Tomi Ylikorpi

The purpose of this thesis was to develop a control method which can reduce oscillation of lateral motion for a pendulum driven spherical robot operating on flat surface. The spherical robot provides a unique mobility and has several applications in surveillance and entertainment.

Controlling a spherical robot is a challenging problem till today due to its nature of kinematics and dynamics. Firstly, its nonholonomic nature prohibits the use of conventional state feedback control laws. Secondly, kinematics of a spherical robot cannot be expressed as a chained-form system to utilize nonholonomic control algorithms. However, various types of nonlinear control algorithms were proposed to settle the problem though none of them provided satisfactory result.

The kinematics and dynamics of the pendulum driven spherical robot was investigated followed by linearization for longitudinal and lateral motions through frequency and state space transformation. Moreover, the controllability of the states of the system was maintained during linearization. A robust self-tuning sliding mode controller which suspends oscillation, maintains desired speed and compensates for unmodeled parameters was developed. The implemented control system consists of control station, prototype robot equipped with on-board microcontroller and sensors, and wireless communication link.

Simulation and experimentation were conducted to test peformance of the control laws in suppressing oscillation and maintaining desired speed of the robot. The robot traveled to the commanded trajectory containing straight line and curve with relatively minimum oscillation at desired speed. Thus, the sliding mode control is an effective controller.

Keywords: Spherical Robot, Pendulum, Sliding Mode Control,Nonholonomic

Contents

List of Figures

Symbols and Abbreviations

ω	vector of quasi-velocities
V	potential energy of outer shell and gyro case
m	total mass of the robot
τ	a torque control input vector
λ	constraint force vector
q	generalized coordinates
$B(q)$	input transformation matrix
$A(q)$	matrix related with nonholonomic constraints
$\bar{T}$	kinetic energy
$M(q)$	mass matrix
$C(q,\dot{q})$	coriolis and centrifugal force vector
$G(q)$	gravitational force vector
N_I	generalized force
n	number of the generalized coordinate q_j
ω_i	quasi-velocities i=1...5
ψ,θ,ϕ	the pose of the robot in ZXZ Euler angle
r	radius of the robot
x, y	geometric center of the robot
l_{x1},l_{y1}	projections of length L on axes x_1,and y_1
$\dot{u}$	transverse angular velocity (roll)
$\dot{v}$	forward angular velocity (pitch)
$\dot{\phi}$	angular velocity (yaw)
θ_{ref}	desired lean angle
e_u	tracking error
s_u	sliding surface

M_s	mass of spherical shell
M_{gc}	mass of gyro case
M	mass of counter-weight pendulum
J	Moment of inertia of prototype robot
τ_v	Drive torque
τ_u	Tilting torque
τ_m	Motor torque
$\dot{\beta}_{dref}$	desired motor rotation velocity
β_{serr}	Integral term of $\dot{\beta}_{sref} - \dot{\beta}_s$
$\dot{\beta}_s$	motor actual velocity
k_{sp}, k_{si}	proportional gain and integral gain
η	gear ratio
k_i	Integral gain of $\dot{\xi}$
k_j	state feedback gains for j=1,..4
LQR	Linear Quadratic Regulator
PD	Proportional derivative
PI	Proportional integral
SMC	Sliding Mode Control

Chapter 1

Introduction

The spherical robot[1] is one of several types of mobile robots which has attracted many researchers due to the nature of its mobility. It provides significant merits over multi-wheel statically stable robots on account of its high maneuverability, good dynamic stability and low rolling resistance. It has a capacity for omni-directional rolling. Moreover, it is light-weight, compact and has a well-sealed structure. Thus, a spherical robot is very suitable to be used in unfriendly or harsh environments such as outer planets and fields. Furthermore, its highly coupled complex dynamics and non-linear control provide challenges for the control engineering field.

A spherical robot is a nonholonomic system. Hence, traditional state feedback control design methods cannot be applied. The current state of the art focuses mainly on designing mechanical structure with controllable and flexible movement and a practical nonholonomic control method.

Many researchers have proposed different kinds of dynamic modeling and motion control for a spherical robot almost all of which are based on open-loop control. However, the simulation and experiments indicate that it is difficult to achieve precise motion with open-loop control methods and kinematic models (Zhan et al., 2006).

[1] refers to a ball-shaped rolling robot

Different ways to settle this problem have been devised such as smooth time-varying stabilization, discontinuous time-independent stabilization and hybrid stabilization. However, these methods can only fit for nonholonmic[2] systems that can be transformed into a chained form[3] . The spherical robot does not satisfy the necessary condition of differential smoothness required by a chained system. Consequently, control methods proposed for a nonholonomic chained system cannot be applied.

In the previous thesis, (Nagai, 2008) developed a prototype pendulum-driven spherical robot and its dynamic model as well as proportional derivative (PD) controller aimed at reducing sideway oscillation. However, the traditional PD controller has been found inappropriate control scheme for nonholonomic spherical robot, and it does not compensate for modeling uncertainties associated with the simplification of the complex dynamics of the robot.

Hence, the objective of this thesis is to develop a suitable dynamic model and an appropriate controller for suppressing sideway oscillation of the prototype pendulum-driven spherical robot to support manual control on flat-surface operation. These include rebuilding of an existing prototype robot , simulation and software implementation of the control algorithm.

The rest of the thesis is organized in the following manner. Chapter 2 reviews previous studies of typical spherical robots focusing on dynamic modeling and control. It also briefly describes principles of design mechanisms and application of a spherical robots. Chapter 3 describes the modeling of the prototype robot. Chapter 4 is devoted to control algorithm development based on the dynamic model. The design, simulation and implementation of sliding mode controller is briefly described. Chapter 5 explains the control system hardware and software. Chapter 6 presents the simulation and experimental results of the control methods. Chapter 7 consists of concluding remarks and future work based upon the results of the tests and observation.

[2]nonintegrable kinematic constraints and systems with less controllable degree of freedom(DOF) than the actual available number of DOF form nonholonomic system.

[3]kinematics of a system that can be expressed as $\dot{x_1}=u_1$, $\dot{x_2}=u_2$, $\dot{x_i}=x_{i-1}u_1$ for $i \neq 1,2$ with control inputs u_1 and u_2.

Chapter 2

Literature Review

The next subsections discuss some of the most recent works in the dynamic analysis, modeling and control of spherical robots. Moreover, typical locomotion principles of the spherical robots are briefly discussed.

2.1 Construction Methods of a Rolling Robot

This research is not intended to describe the locomotion principles of a spherical robot. However, it is very important to understand locomotion principles in order to model and control a spherical robot. Hence, the basic principles of locomotion for a spherical robot are described in brief.

Most spherical robots are based on the principle of moving the center of gravity of a sphere outside of a contact surface area which causes the robot to fall in that direction. Depending on the application, spherical robots can be designed based upon one of the following seven fundamental principles described below. It is not easy to compare[1] performance of each type of robot since none of them have been optimized for use. Thus, performance factors such as speed,

[1] a detailed comparison of each design mechanism can be found in (Armour and Vincent, 2006)

obstacle traversability, or slope climbing as well as smooth controllability which are critical for autonomous mobility are not maximized (Armour and Vincent, 2006).

2.1.1 Sprung Central Member

This principle of design uses a single driven wheel on a sprung member with a passive wheel at the other end. The central member has two motors for driving and steering driven wheel. A typical sprung Central member spherical robot is Rollo shown in Fig. 2.1.

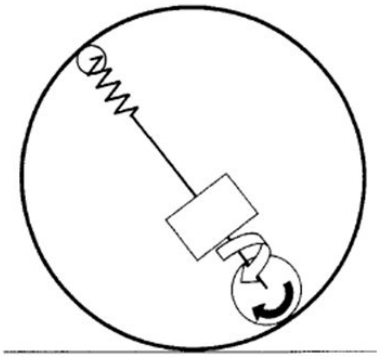

Figure 2.1: design structure of Rollo (reproduced from Halme, 1996)

2.1.2 Car Driven

This design utilizes a small 'car' placed inside a sphere (Fig. 2.2). when the car is driven and steered to the desired direction, the whole sphere can move and steer.

2.1.3 Mobile Masses

The movement of masses along radial axes of a sphere allows to alter the center of gravity of the sphere. The August robot shown in Fig. 2.3 is a typical example

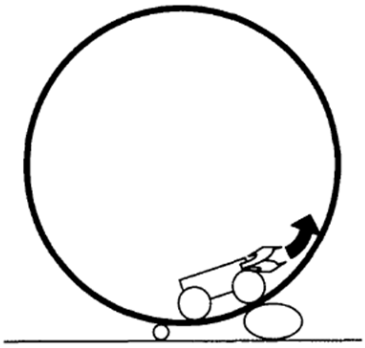

Figure 2.2: Car Driven design, solid arrow shows rolling and open arrow shows steering (reproduced from Rhodri, 2006)

of this type of robot.

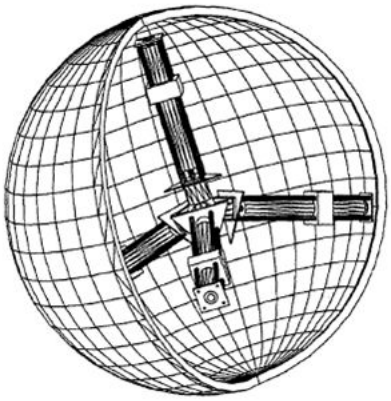

Figure 2.3: August robot structural design (reproduced from Javadi, 2002)

2.1.4 Hemispherical Wheels

The fourth design principle incorporates two hemispheres which are driven individually with a single axle. Kickbot (Fig. 2.4) is a typical hemispherical wheel robot.

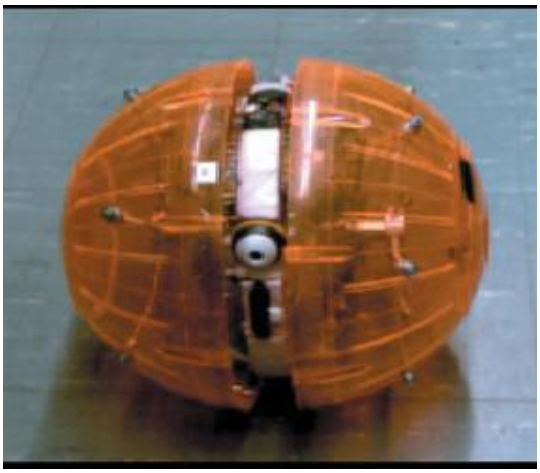

Figure 2.4: Kickbot, a hemispherical wheel robot(reproduced from MIT projects website, *http://www.mit.edu/ cbatten/photos/kickbot-photos.html*)

2.1.5 Gyroscopic Stabilization

In the fifth principle of design, the rolling axis of a wheel is stabilized with a spinning internal gyroscope and a separate motor is used for forward and backward movement. Fig. 2.5 shows the Gyrover robot which works with the principle of gyroscopic stabilization.

Figure 2.5: Gyrover (taken from the gyrover project website, *http://www.cs.cmu.edu/ cyberscout/gyrover.html*)

2.1.6 Ballast Mass Fixed Axis

This design consists of an-off axis mass of a pendulum to change the center of gravity of a robot. The off-axis mass rotates around within a sphere for rolling and the same mass turns sideways to steer towards a desired direction by leaning fixed axis of the sphere. Typical ballast mass fixed axis spherical robots are Marsball (Fig. 2.6) by (Ylikorpi, 2005), Groundbot of Rotundus AB, and Roball of Universite de Shebruke.

Figure 2.6: Marsball (Ylikorpi, 2005)

2.1.7 Ballast Mass Moving Axis

This design is very similar to the ballast mass fixed axis except that the axis is capable of being moved within a sphere. The Rollo (Fig. 2.7) robot by Helsinki University of Technology and the Rover company Ltd. is a typical ballast mass moving axis robot.

The *counter-weight pendulum driven spherical robot*, which is a ballast fixed axis mass, is a popular locomotion mechanism since it is more suitable for autonomy. Therefore, the thesis is devoted to this type of spherical robot.

Figure 2.7: Rollo sphereical robot, 3^{rd} generation developed at TKK (taken from Automation and Systems Technology Department, TKK)

2.2 Applications of Spherical Robots

A spherical robot is used for a number of applications if it is equipped with environmental sensors. These payloads include a wide-angled camera with 360^0 vision, a night vision infrared camera, microphone and loudspeaker, and sensors for radioactivity, gas, humidity, heat and smoke detection, narcotics and explosives.

2.2.1 Surveillance and Reconnaissance

Rotundus, a Swedish company, developed a Groundbot (Fig. 2.8) which is a pendulum-driven spherical robot aimed for applications like

- automated patrolling of large areas
- tele-operated surveillance
- inspection of remote or dangerous sites
- monitoring of explosive gases and etc.

It is widely applicable in airports, power plants, harbors, warehouses, train stations, border security, fuel deposits and stadium event surveillance (Rotundus, 2008).

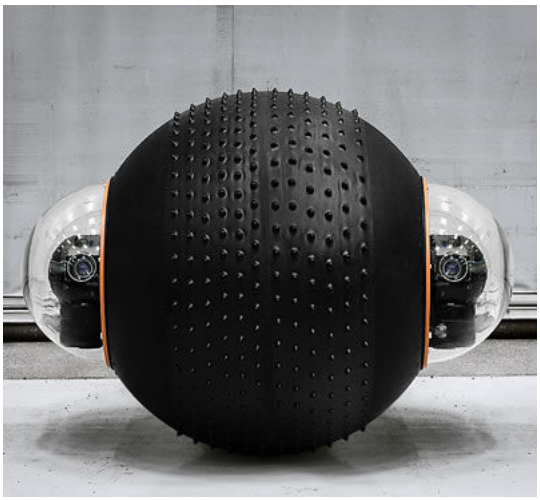

Figure 2.8: Groundbot (Reproduced from Rotundus AB)

Cyclops is another miniature spherical mobile robot designed for remote surveillance and reconnaissance for police and military missions in an urban area. Thus, the nature of mobility and possibility for remote operation of this robot allows humans to survey a dangerous situations from a safe distance (CHEMEL B. and H., 1999).

2.2.2 Planetary Exploration

The outer planet is an extremely hostile environment with significant radiation. Therefore, the electronic circuit requires shielding from the radiation. The spherical robot inherently provides protection for the inner structure consisting of the necessary hardware. Moreover, a spherical robot is lightweight and highly maneuverable. Thus, planetary exploration is one of the attractive application areas of the spherical robots.

2.2.3 Entertainment

The Sony Corporation has developed an entertainment spherical robot called Q.taro (Fig. 2.9), which stands for "Quasi-stable Traveling and Action Robot" shown for the first time back at the Robodex 2002 exhibition of robots in Japan. It is a fully autonomous robot equipped with a number of sensors and it can recognize voice. The robot was developed to foster an emotional connection between humans and robot technology (SONY, 2008).

Figure 2.9: Entertainment robot Q.taro (reproduced from Sony Corporation)

A spherical robot is also used as a playing toy for children. This is because children are used to playing with ball-shaped objects like rubber, plastic or inflatable balls. Hence, a behavior based spherical robot is appealing for children. An experimental test made with Roboll indicates that children become interested and play with it.

Apart from entertainment, a spherical robot is applicable for autism therapy. It will assist a therapist in treatment of autistic children. The robot, an interactive toy is open to direct physical interaction with a child. The behavior of the robot can be programmed by a therapist appropriately for a wide range of child needs (Michuad and Caron, 2002).

2.3 Related Work on Modeling and Motion Control of Spherical Robots

Much research has been performed for decades on the motion control of spherical robots. A spherical robot is considered as a plate-ball system in most modeling. And the research focuses on the controllability and precise motion control for trajectory following. A brief summary of some of the research is presented in (Qiang Zhan, 2008).

The dynamics and controllability of a spherical robot were analyzed a decade ago. For instance, (Li and Canny, 1990) proved the controllability of a sphere with differential geometry and proposed a three-step motion control algorithm to converge its position to the desired values. (Halme et al., 1996) set up kinematic and dynamic models of a spherical mobile robot and analyzed the capabilities of uphill climbing and overrunning obstacles. (Cameron J M, 1997) discussed a kinematic and dynamic modeling of a nonholonomic system and derived a simplified Boltzmann-Hamel equation for both holonomic and nonholonomic systems.

(Bhattacharya and Agrawal, 2000) deduced a first-order mathematical model of a spherical robot from the nonslip constraints and angular momentum conservation and presented simulation and experimental results. (Javadi and Mojabi, 2002) developed the spherical robot called August and established the dynamic model using the Newton method and discussed the motion planning of the robot without feedback control. However, simulations and experiments showed poor accuracy due to the open loop nature of the robot control. (Mukherjee R, 2002) discussed the motion planning and proposed two open-loop control strategies for reconfiguration of a rolling sphere to a desired positions.

More recently (Zhan et al., 2006) discussed the dynamic modeling and trajectory planning of a spherical mobile robot with a simplified Boltzmann-Hamel equation. Also, results of simulations and experiments show that the spherical robot has a strong tendency to oscillate. Thus the authors concluded that open loop control is not a solution for spherical robot motion control and a robust

closed-loop controller and suitable stabilization methods are required.

Based on simulation, (Jia et al., 2008) indicated that the robot can be steered to the position that is desired. More importantly, (Liu et al., 2008a) formulated a kinematic and dynamic model using a constrained generalized Lagrangian Equation. The authors also developed a discontinuous feedback control law using the sliding-mode control method for stabilization. They further demonstrated the effectiveness of the controllers and global asymptotic stabilization and tracking result by simulations. In other research, (Liu et al., 2008b) proposed a proportional integral (PI) controller for drive motion to stabilize velocity and a full-state feedback controller for steering motion and illustrated the effectiveness of the controllers by experiment. Nevertheless, the results indicated only the feasibility of the idea and further improvements are expected in the future.

2.4 Dynamic Analysis, Modeling and Control

The dynamics of a spherical robot is complex and highly non-linear. A number of authors established various types of dynamic models (Vujanovic and Atanackovic, 2003) for spherical robots. Firstly, with the Euler-Lagrangian equation based on the total energy of the system related to external forces; secondly, with the Gibbs-Appell equation which is kinetic energy of acceleration of the system related with generalized applied forces; thirdly, with the Kane equation which is similar to the Euler-Lagrangian equation without concept of virtual displacement and finally, with the simplified Boltzmann-Hamel equation which is a form of Euler-Lagrangian equation in terms of quasi-coordinates.

The spherical robot and its internal units are considered as multibody dynamics in most models. Moreover, the rolling without slipping motion of spherical robots results in a nonholonomic constraint. Accordingly, various control laws are developed based on the model. The following subsections discuss in detail typical modeling and control approaches frequently used by researchers.

2.4.1 Modeling and Control of BHQ-I Robot

BHQ-I[2] is a counter-weight pendulum driven spherical robot similar to GroundBot of Rotundus (Fig. 2.10) which consists of two motors for driving and steering. The rotation of the two motors causes a counter-weight pendulum to rotate about the x and y axes. As a result, the center of gravity of the robot is displaced.

Figure 2.10: structure of GroundBot and BHQ-I (reproduced from Rotundus AB)

The dynamic model is basically derived using kinetic energy associated with generalized and quasi-velocities. These velocities are obtained from the non-slipping constraint condition of the robot. The simplified Boltzman-Hamel equation (2.1) is used to model the dynamic motion of the robot.

$$N_I = \frac{d}{dt}\frac{\partial \bar{T}}{\partial \omega_I} + \sum_{j=1}^{n}\sum_{s=1}^{n}\beta_{sI}\gamma_{sj}\frac{\partial \bar{T}}{\partial \omega_j} - \sum_{j=1}^{n}\beta_{jI}\frac{\partial \bar{T}}{\partial q_j} \tag{2.1}$$

where, ω is the vector of quasi-velocities, $\bar{T}$ is the kinetic energy, N_I is the generalized force, β and γ are the coefficients, I denotes independent quasi-coordinates or related quantities and n is the number of the generalized coor-

[2]Even after extensive searching of Internet resources, the origins of the name is still unclear.

dinate q_j. The chosen quasi-velocities[3] of the robot, expressed in matrix form is

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \omega_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cos\psi & \sin\theta\sin\psi \\ 0 & 0 & 0 & \sin\psi & -\sin\theta\cos\psi \\ 0 & 0 & 1 & 0 & \cos\theta \\ 1 & 0 & 0 & -r\sin\psi & r\sin\theta\cos\psi \\ 0 & 1 & 0 & r\cos\psi & r\sin\theta\sin\psi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \alpha^T \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{2.2}$$

where, ω_1 , ω_2 , ω_3 are the projections of the angular velocities of the robot on axes *i,j,k* of the moving coordinate frame $o'ijk$, which is parallel with fixed frame *oxyz* shown in Fig. 2.11, x and y are geometric center of the robot, r is radius of the robot and ψ, θ, ϕ are the pose of the robot in the ZXZ Euler angles.

From this equation, we can derive the coefficients γ_{ij} from α and β (inverse of α) as in Equation(2.3)

$$\gamma_{ij} = \sum_{k=1}^{5}\sum_{s=1}^{5} \omega_s \beta_{ks} \left(\frac{\partial \alpha_{ij}}{\partial q_k} - \frac{\partial \alpha_{kj}}{\partial q_i}\right) \tag{2.3}$$

Hence, kinetic energy can be expressed by quasi-velocities as

$$\bar{T} = \frac{1}{2} m \left[\frac{7}{5} r^2 (\omega_1^2 + \omega_2^2 + \frac{2}{5} r^2 \omega_3^2 + 2r\omega_2\omega_4 - 2r\omega_1\omega_5 + \omega_4^2 + \omega_5^2\right] \tag{2.4}$$

The dynamic equation is then computed from β, γ, $\bar{T}$ and Equation (2.1)

$$\begin{aligned} \tfrac{7}{5} m r^2 \dot{\omega_1} &= m_1^0 - rV_2 \\ \tfrac{7}{5} m r^2 \dot{\omega_2} &= m_2^0 + rV_1 \\ \tfrac{2}{5} m r^2 \dot{\omega_3} &= m_3^0 \end{aligned} \tag{2.5}$$

where, m_i^0(i=1,2,3) are the projections of the main moment m^0 on axes *i, j, k* of the moving coordinate frame $o'ijk$ shown in Fig. 2.11 on the robot, V_1 *and* V_2 are the projections of force vector V on axes i and j, and m is the mass of the robot.

(Zhan et al., 2006) derived the required input moments of each motor for straight and circular trajectories using Equation (2.5). If the robot moves at a

[3]quasi-velocity is another way of describing velocity of a system and can be linearly related with generalized velocity which is true velocity of the system.

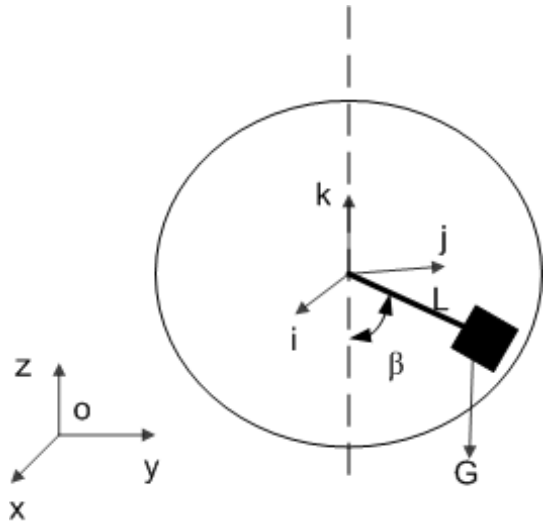

Figure 2.11: Frame assignment for BHQ-I (Zhan et al., 2006)

fixed speed v, the input moments, M_1 and M_2 of the motors are determined: along straight line trajectory

$$\begin{cases} M_1{=}mgl_y = -fr \\ M_2{=}0 \end{cases} \tag{2.6}$$

and along circular trajectory

$$\begin{cases} M_1{=}mgl_{y1} = fr \\ M_2{=}mgl_{x1} = \frac{7mrv^2}{5R} \end{cases} \tag{2.7}$$

where, l_{x1} and l_{y1} are projections of length L on robot centered coordinate axes x_1 and y_1 respectively. The required torque of a motor to drive the robot and its internal mechanism can be selected according to Equations (2.6) and (2.7).

Based on an experiment and simulation, the authors indicated that the robot cannot be precisely controlled with the open-loop control method outlined above. This can be easily seen from the assumption that rotation about the y axis is not considered and decoupled moment equations for each axis are independently described. However, the motion of the robot about its axes is highly coupled and if neglected it requires a compensator.

A related model based on the Boltzman-Hamel equation was later developed for a second prototype model, BHQ-II which is designed for unmanned environment exploration, by (Qiang Zhan, 2008). In this model, a trajectory tracking

controller is designed based on the back-stepping[4] method. Numerical simulations indicate that the controller successfully tracks the desired trajectories asymptotically. However, it was not verified with a practical experiment.

2.4.2 Modeling and Feedback Control of BYQ-III Robot

BYQ-III is a kind of pendulum-driven spherical robot built for research in the School of Automations at the Beijing University of Posts and Telecommunications. The actuation mechanism of this model (Fig. 2.12) consists of two separate actuators: a steer motor which mainly controls the steering motion and a drive motor for forward or backward acceleration (Liu et al., 2008a).

Figure 2.12: Prototype BYQ-III robot (Liu et al., 2008a)

(Liu et al., 2008a) developed a kinematic and dynamic model assuming rigid spherical shell rolling without slipping and a counter-weight pendulum as a particle. The motion of the robot was modeled in the roll(u), yaw(φ) and pitch(v) states neglecting other motions. With these assumptions, the kinematic model

[4]a recursive control law based on the model of the system in which new variables are introduced depending on state variable, controlling parameter and stabilizing functions.

is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} r\sin\varphi & r\cos\varphi\cos u & 0 \\ -r\cos\varphi & -r\sin\varphi\cos u & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\varphi} \end{bmatrix} \tag{2.8}$$

where, φ is heading of the robot, $\dot{u}$ is transverse angular velocity (roll rate), $\dot{v}$ is forward angular velocity (pitch rate), $\dot{\varphi}$ is angular velocity (yaw rate) in the body-fixed frame and (x, y) is position of center of the robot defined in an inertial frame.

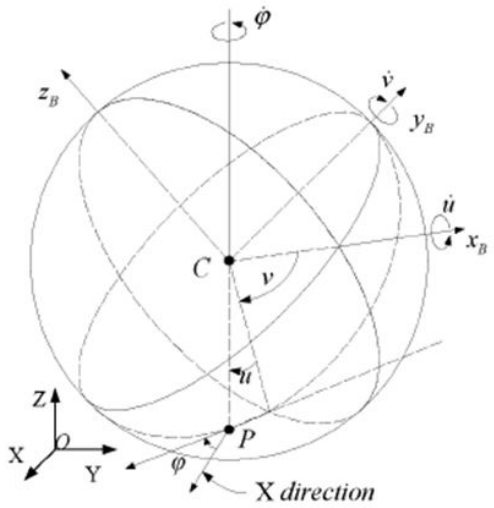

Figure 2.13: Orientation of the shell (Liu et al., 2008a)

The dynamic model is described by the Euler-Lagrange equation under nonholonomic constraints with

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = A^T\lambda(q) + B(q)\tau \tag{2.9}$$

where, q is generalized coordinates, λ is a constraint force vector, τ is a torque control input vector, M(q) is a symmetric and positive definite inertial matrix, $C(q,\dot{q})$ is a centripetal and Coriolis matrix, G(q) is a gravitation vector, B(q) is an input transformation matrix, A(q) is a matrix related with nonholonomic constraints.

With the two control torques, the simplified dynamic equations of the robot

are

$$\begin{aligned}
(mr^2 + I)\ddot{u} &= (mr^2 + I)\cos u\dot{v}\dot{\varphi} + \tau_u \\
(mr^2\cos^2 u + I)\ddot{v} + I\sin u\ddot{\varphi} &= (mr^2 + I)\cos u\dot{u}\dot{\varphi} - \\
\tfrac{1}{2}mr^2\sin 2u\dot{u}\dot{v} &+ \tau_v \\
I\ddot{\varphi} + I\sin u\ddot{v} &= -I\cos u\dot{u}\dot{\varphi}
\end{aligned} \tag{2.10}$$

where, m is total mass of the robot, I is moment of inertia of the robot about its rolling axis and r is radius of the sphere. The lean angle u is coupled with the spinning angle v and steering angle φ at the velocity level through $\dot{v}$ and $\dot{\varphi}$.

Approximate uncertainties are modeled as disturbances during linearization and sliding mode control methods are employed (Liu et al., 2008b). Assuming δu, $\delta\dot{v}$ are sufficiently small, and u=δu, $\dot{v}$=Ω_0+$\delta\dot{v}$, Ω_0 is the nominal value, the linearized and decoupled model can be represented as

$$\delta\ddot{u} = \Omega_0\dot{\varphi} + \Delta f_u + \tau_1 \tag{2.11}$$

$$\delta\ddot{v} = \Delta f_v + \tau_2 \tag{2.12}$$

$$\ddot{\varphi} = -\Omega_0\delta\dot{u} \tag{2.13}$$

where δf_v and δf_u are approximate uncertainties.

The authors have designed closed loop control independently for controlling the velocity $\dot{v}$ to the nominal value Ω_0. Accordingly, τ_2 is designed to robustly stabilize the origin $\delta\dot{v} = 0$ as

$$\tau_2 = -k_{v1}\delta\dot{v} - k_v.sign(\lambda_v\delta\dot{v}) \tag{2.14}$$

where $-k_{v1} > 0$, k_v>sup$|\delta f_v|$, and λ_v>0.

The sliding mode control with sliding surface, s_v=$\lambda_v\delta\dot{v}$ is proved to stabilize the origin using the Lyapunov function[5]. For position control, torque control

[5]Lyapunov method determines the stability of an equilibrium point without solving the state equation using a continuously differentiable function called Lyapunov function.

input τ_1 was chosen from Equation (2.11) as

$$\begin{aligned}
\tau_1 = & \; -k_{u1}\dot{e_u} - k_{u2}e_u - \beta(\delta u).sign(s_u) - \Omega\dot{\varphi} \\
e_u = & \; \delta u - u_r \\
s_u = & \; \dot{e_u} + \lambda_u e_u \\
\beta(\delta u) = & \; a|\delta\dot{u}| + b|\delta u| + \eta|R| + sup(\Delta f_u) \\
R = & \; -\delta\ddot{u_r} - (k_{u1} + \lambda_u)\delta\dot{u_r} - k_{u2}\delta u_r
\end{aligned} \tag{2.15}$$

where a, b, η, k_{u1} and k_{u2} are positive parameters which satisfy the conditions: $\lambda_u - k_{u1} \leq a$, $-k_{u2} \leq b$ and $1 < \eta$.

(Liu et al., 2008a) showed that the control input stabilizes the sliding surface with the presence of the disturbance Δf_u in the system. Position-tracking errors converge to zero and the heading-direction is bounded. Furthermore, the authors developed a line following controller for tracking a desired line based on the derivative of the curvature of the path. The path curvature of the contact point can be expressed by a steering function as in (2.16) in terms of the lean angle. Thus, it is possible to indirectly control the path curvature to steer the robot by changing its lean angle.

$$k(t) = \frac{1}{\rho(t)} = \frac{\tan u(t)}{r} \tag{2.16}$$

where $\rho(t)$ is the radius of the curvature from the center of rotation.

Position control law can stabilize the robot to a predefined lean angle which corresponds to the desired path curvature.

The proposed sliding-mode controllers to stabilize the tracking errors in lean angle and spinning velocity which indirectly stabilize the desired path curvature were effective according to numerical simulations (Liu et al., 2008b). However, the results of the simulation which considered the non-linear and coupled model of the robot may not perform well in the real world without modification. This is because the assumptions made may not hold in a real dynamic environment due to a number of unmodeled errors and disturbances. However, sliding mode control is a robust controller which can compensate unmodeled errors if it is implemented with proper controller coefficients and a sliding surface.

2.4.3 State Feedback Linearization

The BYQ-III robot described in Section (2.2.2) was modeled based on the Euler-Lagrange equation by considering the robot as a multi-body system composed of two rigid links and regarding the pendulum as a particle. Additionally, a state feedback controller was designed. The basic assumptions in deriving the dynamic model were

- rotational axis of gyro case is parallel to one of the principal axes of inertia of gyro case
- no slipping on the horizontal plane
- outer shell is a spherical one that is completely symmetrical
- effects of various types of friction are modeled as pure viscous damping

The equations for kinetic energy, T and potential energy, V for a system consisting of a spherical shell and inner driving mechanism is described with the Euler-Lagrange equation

$$\frac{d}{dt}\frac{\partial L}{\partial q} - \frac{\partial L}{\partial q} + A^T\lambda = Bu \tag{2.17}$$

and L=T-V is total energy computed for the simplified model in Fig. 2.14 as

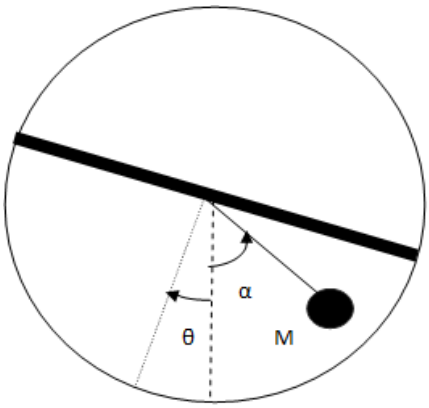

Figure 2.14: Simplified model of the robot (reproduced from Liu and Wang, 2008)

$$L = \frac{a\dot{\theta}^2}{2} - b\dot{\theta}\dot{\alpha}\cos\alpha + \frac{c\dot{\alpha}^2}{2} - d\cos\alpha - V - d \tag{2.18}$$

where, $a = M_s r^2 + M_{gc} r^2 + Mr^2 + I_s + I_{gc}$, $b = Mrl$, $c = Ml^2$, $d = Mgl$, V is a constant potential energy due to symmetry of the spherical shell and gyro case, M_s is mass of the shell, M is mass of the counter-weight pendulum, M_{gc} is mass of the gyro case, I_s is moment of inertia of the shell, I_{gc} is mass of the shell, r is radius of the sphere and l is length between centers of the shell and the pendulum.

Computing Lagrangian derivatives in (2.17), the nonlinear equation of motion can be expressed as

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \tag{2.19}$$

where the generalized coordinates for this system are

$$q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T = \begin{bmatrix} \theta & \alpha \end{bmatrix}^T \tag{2.20}$$

The standard nonlinear state space form of Equation (2.19) is

$$\dot{x} = f(x) + g(x)u \tag{2.21}$$

where

$$x = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}. \tag{2.22}$$

Although the rolling and steering motions in their respective planes are highly coupled to each other, they can be decoupled under the assumption that the rotation velocity is low (Liu et al., 2008b). Thus, it is feasible to design a linear feedback law for the system to control the lean angle and rolling speed of the robot. Accordingly, the rotation velocity of the motor $\dot{\beta}_{dref}$ can be stabilized by designing a linear state feedback according to Equation (2.23)

$$u_\beta = k_{dp}(\dot{\beta}_{dref} - \dot{\beta}_d) + k_{di}\beta_{derr} \tag{2.23}$$

where, β_{derr} is the integral term of $(\dot{\beta}_{dref} - \dot{\beta}_d)$.

A separate control law is used to stabilize the desired lean angle (Liu et al., 2008b). It consists of firstly an inner loop which feeds the actual velocity of a steer motor back into a PI controller; secondly, a mid loop Linear Quadratic Regulator (LQR) that uses full-state feedback; and finally, an outer loop integral controller which tracks the desired lean angle and automatically compensates for frictional torques that must be overcome to achieve tracking as shown in Fig. 2.15 below.

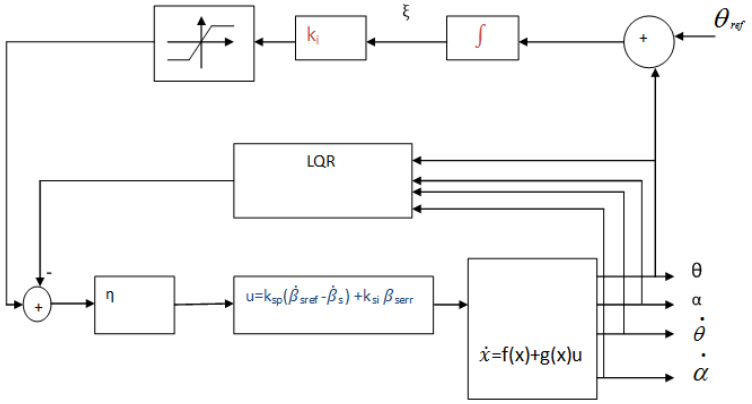

Figure 2.15: structure of lean angle controller (reproduced from Liu and Wang, 2008)

The augmented state vector which arises from the integral terms added to the system is

$$x_a = \begin{bmatrix} x^T & \beta_{serr} & \xi \end{bmatrix}^T \tag{2.24}$$

Hence, the closed loop standard nonlinear state space form can be rewritten as

$$\dot{x_a} = \begin{bmatrix} f(x) + g(x)(k_{sp}(\dot{\beta}_{sref} - \dot{\beta}_s) + k_{si}\beta_{serr} \\ \dot{\beta}_{serr} - \dot{\beta}_s \\ \theta_{ref} - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x \end{bmatrix} \equiv f_a(x_a, \theta_{ref}) \tag{2.25}$$

Linearizing this equation about its equilibrium point $\dot{x_a}$=0, a controllable state was obtained in

$$\dot{x_a} = \underbrace{\frac{\partial f_a}{\partial x_a}|_{x_a=0,\theta_{ref}=0}}_{A} x_a + \underbrace{\frac{\partial f_a}{\partial \theta_{ref}}|_{x_a=0,\theta_{ref}=0}}_{B} \theta_{ref} \tag{2.26}$$

Although the authors deduced a smooth stabilizing controller from the controllability of the pairs A and B, the nonlinear state equation in Equation (2.25) does not provide a controllable state when it is linearized about its equilibrium according to Equations (2.27) and (2.28). Thus, nonholonomicity of the robot prevents an approximate controllable linear state equation.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{2bd+cd}{b^2-ac} & 0 & 0 & 0 & 0 \\ 0 & \frac{bd}{b^2-ac} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.27}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

$$rank \begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix} \neq 6 \tag{2.28}$$

However, the authors demonstrated the effectiveness of the controllers with an experiment although they pointed out that the controller should be improved to obtain better dynamic performance of the robot in future (Liu et al., 2008b) .

Chapter 3

Modeling of the Prototype Spherical Robot

In this chapter, kinematic and dynamic analysis, and modeling of the prototype spherical robot (Fig. 3.1) developed at Automation and Systems Technology department, TKK is discussed. The robot is equipped with two functionally different motors. A DC motor provides longitudinal forward and backward motion on a plane while a servo motor provides steering motion to the robot to a desired direction in another plane. The servo motor is loaded with a counter-weight pendulum suspended with an arm to provide torque for displacement of the center of gravity of the robot to achieve the required longitudinal and lateral motions.

3.1 Steering Torque

The servo motor can steer the robot by leaning in the desired direction. Thus, leaning torque is indirectly provided by the servo motor. When the servo motor lifts a counter-weight pendulum in either direction, the gravitational force due to the mass of the pendulum exerts a torque (Fig. 3.2). It is this torque that tilts the robot to a specified orientation.

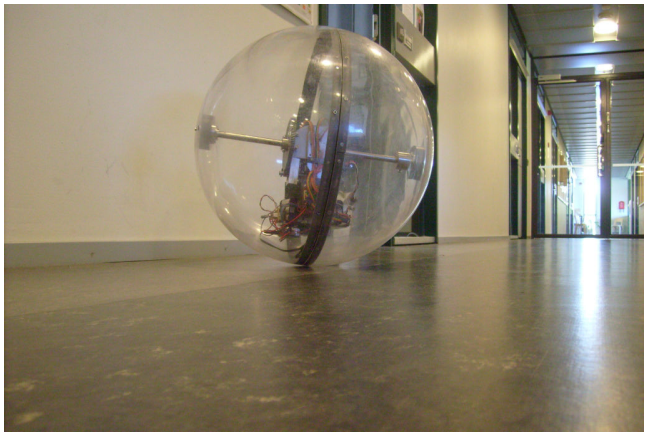

Figure 3.1: Prototype pendulum driven spherical robot

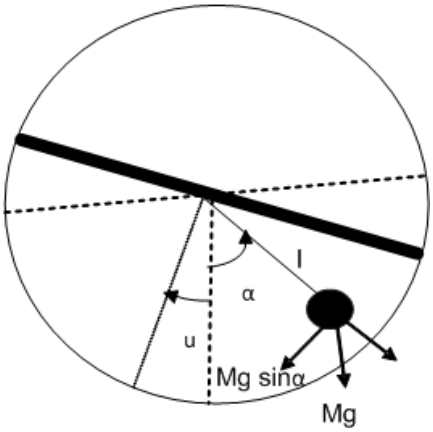

Figure 3.2: Gravitational torque which leans the robot for steering motion

The lean torque exerted due to steering with servo motor is

$$\tau_u = Mglsin\alpha \tag{3.1}$$

where, α is tilt angle of the pendulum from the vertical, l is the length of the arm from the center of the robot to that of the pendulum, M is mass of the pendulum and g is gravitational acceleration.

Hence, the lean angle control for steering can be achieved with servo torque. Moreover, the path curvature of the robot which is the inverse of the radius of the curvature ρ, is controlled by controlling the lean angle of the robot according to Equation (3.2). There exist a centrifugal force and gyroscopic effect which balance steering force to stabilize the robot in a curvilinear motion. However, the centrifugal force acting on the robot is very small for a low angular rate of

veering and small radius of curvature[1], and gyroscopic effect is negligible for a low speed of the robot.

$$\frac{1}{\rho} = \frac{\tan u}{r} \tag{3.2}$$

3.2 Dynamic Analysis of Longitudinal and Lateral motions

The accuracy of modeling actual dynamics of the ball shaped robot determines accuracy of control method. Moreover, it is desirable to model the complete motion mathematically in such a way that can be utilized in control algorithms. In general, there are two ways of viewing the dynamics of the spherical robot depending on the state variables used in the control algorithm. Firstly, a complete configuration of a spherical robot can be described with six independent coordinates (position and orientation) by considering the total dynamics of internal mechanism as an entity. Secondly, the spherical robot and its internal mechanism are modeled as a multi-body system. In the following subsection, the former is considered for the analysis of the prototype robot.

3.2.1 Dynamic Model of the Robot as a Spherical Body

In this model, the total motion of the robot is considered in a such a way that the inner mechanism and the mass of the pendulum is included in a 3-D model of the robot as shown in Fig. 3.3. This modeling approach is preferred over multi-body system for two reasons. Firstly, it clearly portrays Euler angles and Euler rates of the robot associated with lateral and longitudinal motions for advanced control development. Secondly, the actuator which drives the pendulum has built-in control system to control angular position and it drives the pendulum at constant speed. Thus, the states of the pendulum have been already controlled. However, the effects of the pendulum dynamics are incorporated with entire

[1]note that the motion of the robot is rotational motion without translational velocity.

motion of the robot through virtual torque about longitudinal and lateral axes at the center of the robot.

The robot is assumed to roll without slipping. Hence, the velocity of the robot, **v** at the instant in contact with the surface at p_c is zero.

$$\mathbf{v} + \omega \times \mathbf{r} = 0 \tag{3.3}$$

where, **r** is the radius vector whose magnitude is the radius of the sphere. The angular velocity, ω of the robot with respect to an inertial frame can be computed by transformation of the body frame coordinates as

$$\omega = \dot{u}i + (\dot{\varphi}\sin u + \dot{v})j + (\dot{\varphi}\cos u)k \tag{3.4}$$

where, (i,j,k) are unit vectors of the body-fixed frame.

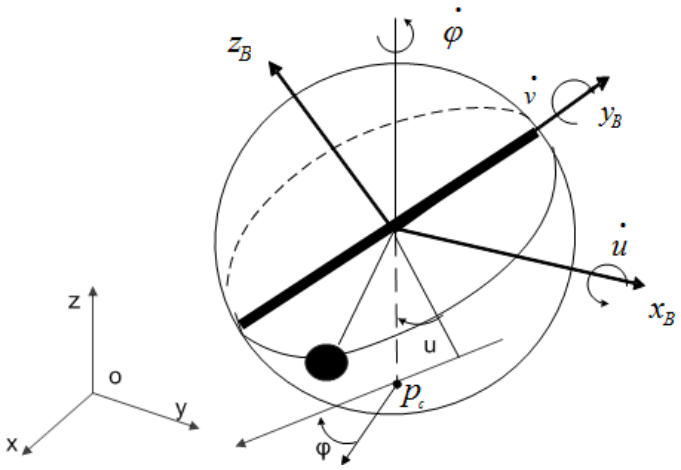

Figure 3.3: Configuration of the prototype robot in body and inertial frames

The constraint condition in Equation (3.3) provides Equation (3.5) in an inertial frame as described in Equation (2.8).

$$\begin{aligned} \dot{x} &= r\sin\varphi\dot{u} + r\cos\varphi\cos u\dot{v} \\ \dot{y} &= r\cos\varphi\dot{u} - r\sin\varphi\cos u\dot{v} \end{aligned} \tag{3.5}$$

The generalized coordinates are defined as u, v and φ. Hence, the Lagrange equation is

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{v}} - \frac{\partial L}{\partial v} = n\tau_m - T_f \tag{3.6}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{u}} - \frac{\partial L}{\partial u} = \tau_u \tag{3.7}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} - \frac{\partial L}{\partial \varphi} = 0 \tag{3.8}$$

where, $\dot{x}$ and $\dot{y}$ are velocities in x and y directions respectively, ω is angular velocity of the robot, τ_m is torque provided by DC drive motor, T_f is viscous friction between the spherical shell and the surface, n is gear ratio, φ is heading angle (yaw) of the robot, u is lean angle (roll) of the robot which is the sideway motion with respect to the progression, v is spinning angle (pitch) of the robot in the direction of the progression and r is radius of the spherical shell.

The friction force between the robot and the surface is assumed to be Coulomb and viscous friction. The Coulomb friction is constant and viscous friction depends on the velocity of the robot, $\dot{v}$. Thus, the total friction force is

$$T_f = k_f \dot{v} r + T_c \tag{3.9}$$

where k_f is the viscous friction coefficient of the surface
The Lagrangian parameters are given by

$$\begin{aligned} L=&\ K + T - V \\ K=&\ \tfrac{1}{2} m(\dot{x}^2 + \dot{y}^2) \\ T=&\ \tfrac{1}{2} J(\dot{u}^2 + \dot{v}^2 + \dot{\varphi}^2 + 2\sin u \dot{\varphi} \dot{v}) \\ V =&\ 0 \end{aligned} \tag{3.10}$$

where, L is total energy, K is translational kinetic energy of the robot, T rotational kinetic energy of the robot, V is potential energy of the robot from the center of the sphere, m is total mass of the robot and J is moment of inertia of the robot.

From Equations (3.6) - (3.10), the modified equation of motion for the prototype robot is

$$\begin{aligned} &(mr^2 \cos^2 u + J)\ddot{v} + J \sin u \ddot{\varphi} = \ (mr^2 + J)\cos u \dot{u} \dot{\varphi} - \\ &\tfrac{1}{2} mr^2 \sin 2u \dot{u} \dot{v} - k_f \dot{v} r - T_c + \tau_v \\ &(mr^2 + J)\ddot{u} = \ (mr^2 + J)\cos u \dot{v} \dot{\varphi} + \tau_u \\ &J\ddot{\varphi} + J \sin u \ddot{v} = \ - J \cos u \dot{u} \dot{v} \end{aligned} \tag{3.11}$$

where, τ_v is applied torque to the robot for driving and τ_u is applied lean torque for steering.

Equation (3.11) completely describes the 3-D model of the prototype robot by revealing its yaw, roll and pitch motions.

Chapter 4

Control Algorithm Development

The dynamic model of the prototype robot described in the previous chapter is not suitable for controller design due to the coupled and nonlinear dynamics. Hence, it is required to approximate to benefit from the model for control algorithm development.

The most common approach to reduce a complex dynamic model into a usable form for control is through linearization. However, an approximate linearization of nonholonomic systems causes loss of controllability. Hence, it is necessary to maintain controllability of the system during linear approximation. A different approach from previous studies is presented to tackle such problem in this thesis as outlined in Fig. 4.1 through transformation from time domain to frequency domain and vice versa. Firstly, coupled nonlinear dynamics is partially

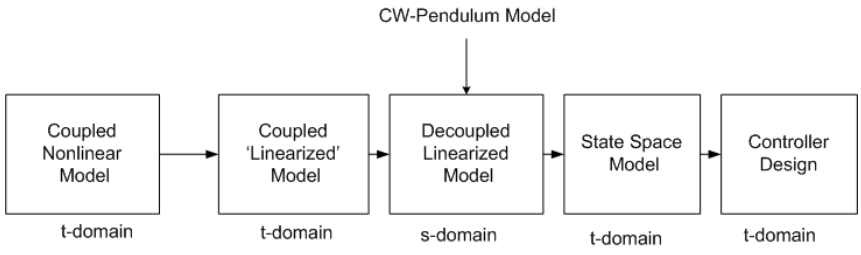

Figure 4.1: Sequence of operations for controller design

linearized for practically small roll angle. Secondly, this coupled linearized

model is described using the Laplace transform which results in decoupled linear model in frequency domain and finally, state space models of longitudinal and lateral motions are developed to design the controllers.

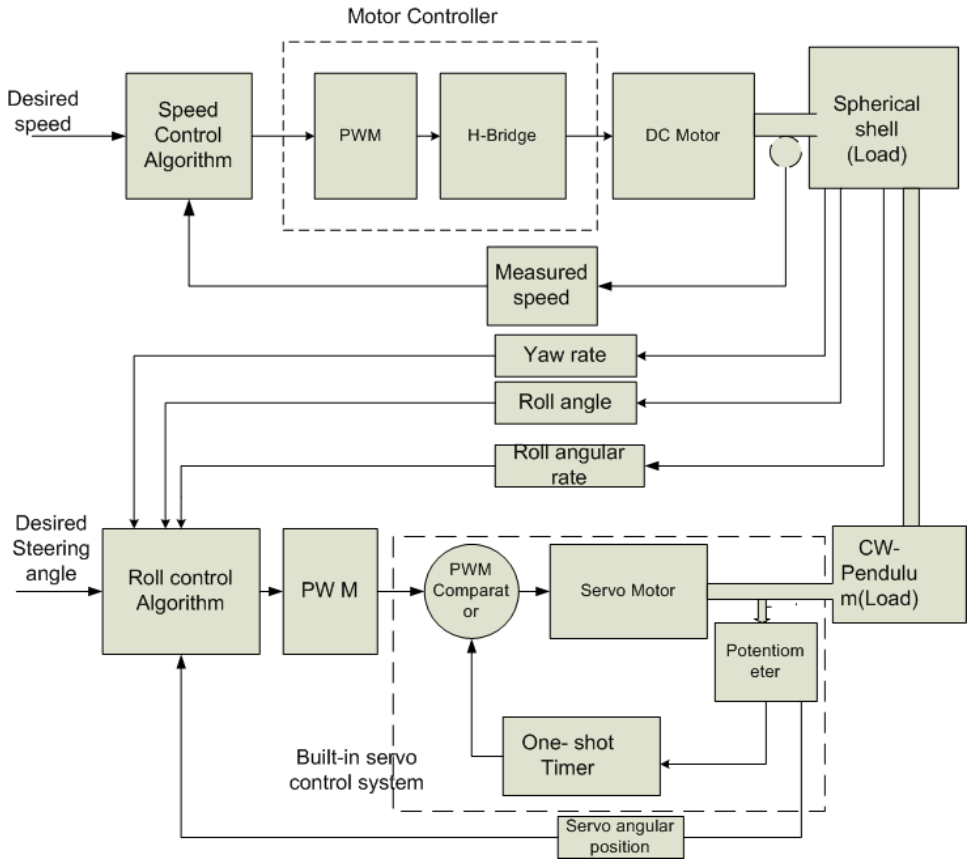

Figure 4.2: Control system architecture

The control system architecture of both longitudinal and lateral motions is shown in Fig. 4.2. Here, the counter-weight pendulum (CW-pendulum) and spherical shell are described as separate loads of driving and steering motors for simplicity of representation. However, the DC motor drives both the spherical shell and pendulum.

4.1 Approximated Linear Model with Compensation

The highly coupled dynamics of the prototype robot is linearized according to (Liu et al., 2008a) and modified. The coupling terms are eliminated from the equation and compensated with an uncertainty model Δf_u and Δf_v.

For a sufficiently small roll angle δu

$$\begin{aligned} \cos \delta u &\approx 1 \\ \sin \delta u &\approx \delta u \end{aligned} \tag{4.1}$$

Hence, Equations in (3.11) can be rewritten as

$$\delta \ddot{v} = \delta \dot{u} \dot{\varphi} + \frac{\tau_v}{mr^2 + J} - k_f \dot{v} r + T_c = \Delta f_v + \tau_2 - k_f \dot{v} r + T_c \tag{4.2}$$

$$\delta \ddot{u} = \Omega_0 \dot{\varphi} + \delta \dot{v} \dot{\varphi} + \frac{\tau_u}{mr^2 + J} = \Omega_0 \dot{\varphi} + \Delta f_u + \tau_1 \tag{4.3}$$

$$\ddot{\varphi} = -\Omega_0 \delta \dot{u} \tag{4.4}$$

$\Delta f_u = \dot{\varphi} \delta \dot{v}$ and $\Delta f_v = \dot{\varphi} \delta \dot{u}$ are variable model parameters whose values can not be predetermined. However, the robust sliding mode controllers compensate for the variation of these parameters during motion. $\delta \dot{v}$ is a small velocity deviation from the nominal speed Ω_0.

4.1.1 Longitudinal Motion and Driving Motor Model

The DC motor which drives the sphere forward and backward is supplied with voltage and generates torque according to Equation (4.5).

$$\begin{aligned} E &= IR + k_e \omega_m \\ \tau_m &= k_t I = \tfrac{1}{R}(k_t E - k_e k_t \omega_m) \end{aligned} \tag{4.5}$$

where, E is supply voltage to the DC motor, k_e is speed constant, k_t is torque constant, R is resistance of motor winding, I Current flowing to the motor, ω_m is Speed of the DC motor.

The shaft of the drive motor is coupled with a torque transmission shaft via two stage gears(Fig. 4.3). The first stage has a driving gear (pinion) mounted on the motor with N_1 teeth, and a driven gear mounted on the intermediate shaft with N_2 teeth. The second stage has a driving gear mounted on the other end of the intermediate shaft with N_3 teeth and the driven gear mounted on rolling axis with N_4 teeth. Assuming that the loss of power through gears is negligible, the resultant gear ratio is calculated.

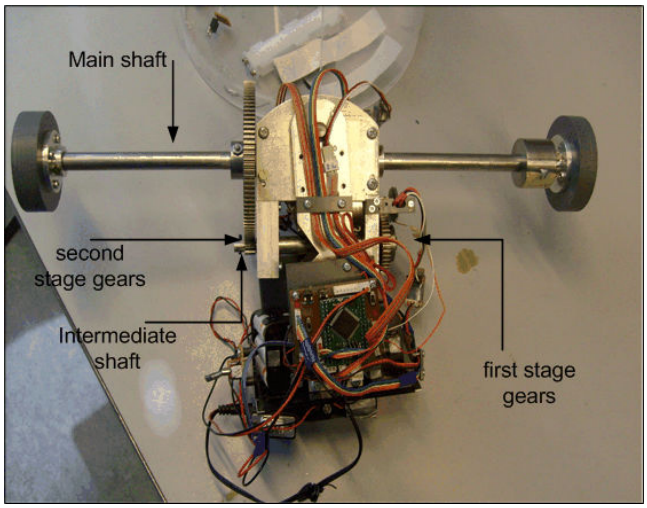

Figure 4.3: Driving torque gear system

$$T_2 = \frac{N_2}{N_1} T_m \tag{4.6}$$

$$\tau_v = \frac{N_4}{N_3} T_2 \tag{4.7}$$

Combining Equations (4.6) and (4.7), the load torque provided by the DC motor via gear system is

$$\tau_v = \frac{N_4 N_2}{N_3 N_1} T_m \tag{4.8}$$

Hence, the total gear ratio is

$$n = \frac{N_4 N_2}{N_3 N_1} \tag{4.9}$$

Transforming the motor torque to the load through gear ratio,

$$\tau_v = nT_m = \frac{nk_tE}{R} - \frac{n^2k_ek_t\dot{v}}{R} \tag{4.10}$$

Laplace transforming Equation (4.2), speed to torque model is

$$sV(s)(s + k_fr) = \frac{\tau_v}{mr^2 + J} + T_c + \Delta f_v \tag{4.11}$$

Direct torque control is not suitable for manual operation of the robot, hence the speed of the motor and resulting speed of the robot is controlled via the supply voltage according to the model shown in Fig. 4.4.

$$sV(s)(s + k_fr + \frac{n^2k_ek_t}{R(mr^2 + J)}) = \Delta f_v + T_c + \frac{nk_t}{R(mr^2 + J)}E \tag{4.12}$$

The ideal transfer function of speed of the longitudinal motion to torque and supply voltage respectively is

$$\frac{\omega_v(s)}{\tau_v(s)} = \frac{1}{mr^2 + J}\frac{1}{s + k_fr} \tag{4.13}$$

$$\frac{\omega_v(s)}{E(s)} = \frac{nk_t}{R(mr^2 + J)}\frac{1}{s + (k_fr + \frac{n^2k_ek_t}{R(mr^2+J)})} \tag{4.14}$$

where, $\omega_v(s) = sV(s)$ is the speed of the robot.

Equation (4.14) indicates that forward and backward driving motion is stable through voltage control. The extent of stability is determined by physical parameters of the DC motor and sphere which determine the positions of the poles of the system. .

4.1.2 Lateral Motion Model

The steering of the robot can be modeled approximately with the linearized Equations (4.3) and (4.4). Hence, the Laplace transformation of these equations are given by

$$s^2\delta u(s) = \Omega_0 s\varphi(s) + \Delta f_u + \frac{\tau_u}{mr^2 + J} \tag{4.15}$$

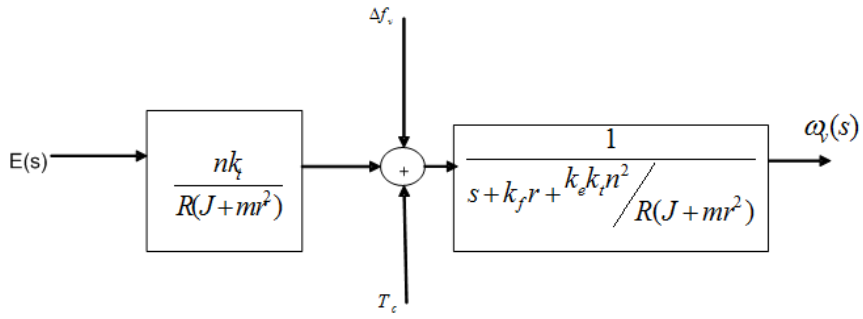

Figure 4.4: Block diagram of the relationship between control voltage and speed of the robot

$$s^2\varphi(s) = -\Omega_0 s\delta u(s) \tag{4.16}$$

Combining Equations (4.15) and (4.16), the roll angle to torque transfer function is given by

$$\delta u(s)(s^2 + \Omega_0^2) = \Delta f_u + \frac{\tau_u}{mr^2 + J} \tag{4.17}$$

The torque required to lean the robot by steering the pendulum mass can be approximated for a small steering angle α, where $\sin\alpha \approx \alpha$ as

$$\tau_u = Mgl\alpha \tag{4.18}$$

Hence, the transfer function of the steering angle of the pendulum to the lean angle of the robot is

$$\delta u(s) = \frac{\tau_u}{mr^2 + J}\frac{1}{s^2 + \Omega_0^2} + \frac{\Delta f_u}{s^2 + \Omega_0^2} \tag{4.19}$$

And the input output relation is shown in block diagram in Fig. 4.5. The poles

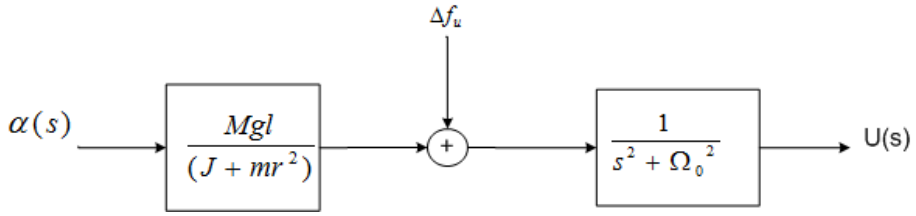

Figure 4.5: Transfer function of lean angle and titlt angle of the pendulum

are

$$s = \pm j\Omega_0 \tag{4.20}$$

Thus, the poles of the transfer function

$$\frac{\delta u(s)}{\alpha(s)}$$

are on the imaginary axis (Fig. 4.6) which indicates that the lateral motion of the robot is oscillatory. Due to simplification of the model, the lateral sideway motion is in undamped oscillation. However, the model agrees with very slowly damping oscillation model (Nagai, 2008) with poles at $s = a \pm j\Omega_0$ where a is negligible value.

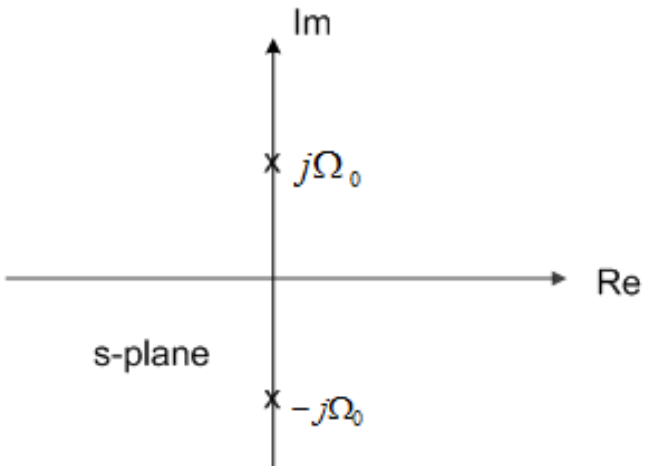

Figure 4.6: Poles of Lateral motion

Therefore, active and/or passive suppression of sideway oscillation is required for smooth operation of the robot. A damper inside the sphere can be used to reduce the oscillation although it causes hardware changes of the system. Alternatively, an external damper can be used on the outer rolling surface of the robot at the cost of increasing rolling resistance as shown in Fig. 4.7. However, an active oscillation suppression is mandatory although a passive damper helps reduce the oscillation. Thus, this research focuses on active control methods such as sliding mode control in the remaining sections of this chapter.

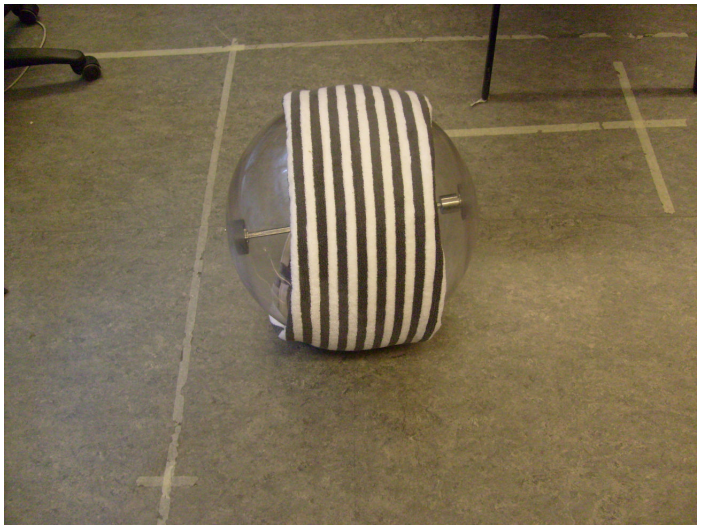

Figure 4.7: Proposed passive sideway oscillation Model

4.2 Sliding Mode Control Design

Sliding mode control(SMC) is a standard robust control approach to tackle the parametric and modeling uncertainties of a nonlinear system. A typical structure of SMC is composed of two parts: nominal part, similar to feedback control and nonlinear part, additional terms aimed at dealing with uncertainty.

In sliding mode, the system is driven to a stable manifold and slide to equilibrium. The objective of switching control law is to drive the nonlinear system state trajectory onto a designer-chosen surface in the state space. The system state trajectory is maintained on a sliding surface for the subsequent time. When the state trajectory is above the sliding surface, a feedback path has one gain and a different gain when the trajectory goes below the surface. Ideally, the switched control maintains the system state trajectory on the surface for all the subsequent time and the state trajectory slides along this surface. However, there exist drawbacks of sliding mode control: its sensitivity to controller parameters and the tendency to oscillate due to switching around the sliding surface.

The system is represented in a state space to design the sliding mode controller. In addition, the stability of an equilibrium point is determined by the Lyapunov

method without solving the state equation. Moreover, the sliding mode control law should take the shortest possible time during the reaching phase and it should not generate any chattering[1] during the sliding phase. Accordingly, the speed and roll angle controller are separately designed and implemented by reducing the inherent drawbacks arising from traditional sliding mode control.

4.2.1 Speed controller

The state space model of longitudinal motion which is convenient to design sliding mode control is shown in Equation (4.21). The controllability of the system and stability of the sliding mode speed controller are important factors that determine the effectiveness of the sliding mode controller design.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -k_f r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau_2 \tag{4.21}$$

where, x_1 and x_2 are state variables representing the pitch angle and velocity of the robot respectively and τ_2 is control input (torque scaled by the inertia of the robot) of the longitudinal motion.

The controllability matrix R, given by Equation (4.22) is full rank (rank =2), hence longitudinal motion is controllable.

$$R = \begin{bmatrix} 0 & 1 \\ 1 & -k_f r \end{bmatrix} \tag{4.22}$$

x_1 is stable if $\dot{x_1} = -k_{v1} x_1$, $k_{v1} > 0$. Defining stable manifold as

$$s = x_2 + k_{v1} x_1, \tag{4.23}$$

$$\dot{x_1} = x_2 = -k_{v1} x_1 + s \tag{4.24}$$

is stable if s= 0. The time derivative of s is given as

$$\dot{s} = \dot{x_2} + k_{v1} \dot{x_1} = -k_f r x_2 + \tau_2 + k_{v1} x_2 \tag{4.25}$$

[1] phenomenon of a high frequency oscillation

The stability can be evaluated using the Lyapunov function and its derivative as

$$\begin{aligned} V &= \tfrac{1}{2}s^2 \\ \dot{V} &= s\dot{s} = s(-k_f r x_2 + \tau_2 + k_{v1} x_2) \end{aligned} \tag{4.26}$$

$\dot{V}$ is negative definite if

$$-k_f r x_2 + \tau_2 + k_{v1} x_2 \begin{cases} < 0 & \text{if } s > 0 \\ = 0 & \text{if } s = 0 \\ > 0 & \text{if } s < 0 \end{cases}$$

According to Lyapunov criteria, stability is assured by using the control law

$$\tau_2 = -(k_f r + k_{v1})x_2 - k_v sign(s) \tag{4.27}$$

The sliding mode controller in Equation (4.27) is used to stabilize the speed of the prototype robot to the desired nominal speed. The complete longitudinal motion model with sliding mode speed controller is shown in Fig. 4.8. A smooth stabilizing controller without chattering is achieved with $k_{v1} = 1$ and $k_v = 1$ through experimentation.

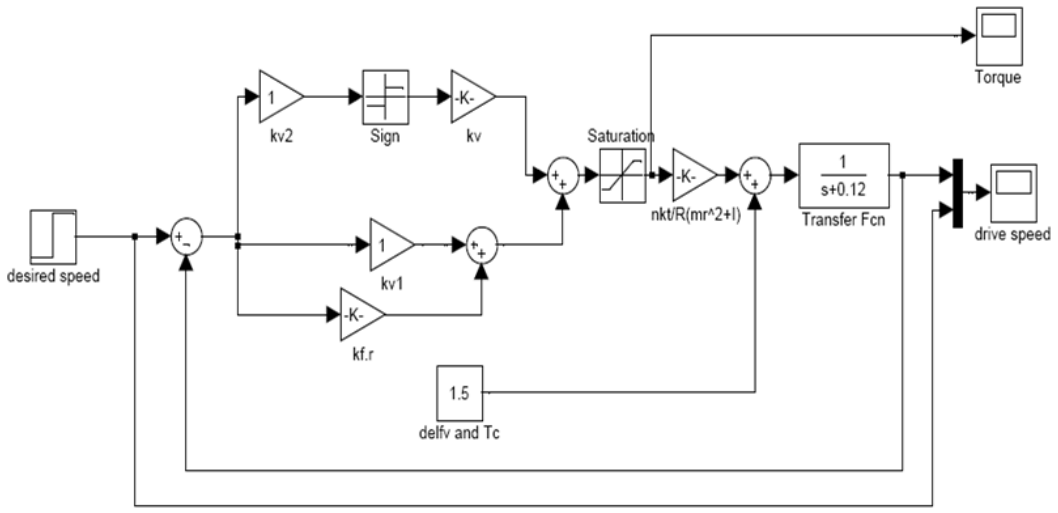

Figure 4.8: Speed control simulation Model

4.2.2 Roll angle controller

A control law which can suppress the oscillation of a roll angle of lateral motion is developed based on sliding mode control, and the results are discussed in Chapter 6. The concept of the control algorithm is partly adapted from (Liu et al., 2008a) and modified for a real time application.

The state space representation of linearized model of lateral motion is

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\Omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau_1 \tag{4.28}$$

where, x_1 and x_2 are roll angle and roll angle rate of the robot respectively and τ_1 is control input of lateral motion. From the state matrix and input constant vector of Equation (4.28), the controllability matrix

$$R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.29}$$

is a full rank. Thus, the lateral motion is controllable.

From an approximated model of the prototype robot, it can be deduced that a proportional derivative controller is required to eliminate the oscillation of lateral motion. This can be achieved with sliding mode control which also compensates for modeling errors with its nonlinear term. Hence, the sliding mode control is designed to eliminate the lateral oscillation of the robot.

The state variables of the model are modified as

$$\begin{aligned} x_1 &= e_u = \delta u - u_r \\ x_2 &= \dot{e_u} = \delta \dot{u} - \delta \dot{u_r} \end{aligned} \tag{4.30}$$

where u_r is the reference roll angle.
e_u is stable if $\dot{e_u} = -k_{u1} e_u$, $k_{u1} > 0$. Hence, the convergence of the lateral oscillation can be achieved with a sliding surface

$$s_u = \dot{e_u} + k_{u1} e_u \tag{4.31}$$

With the Lyapunov function,

$$V(s_u) = \frac{1}{2}s_u^2 \tag{4.32}$$

$$\dot{V} = \dot{s_u}s_u = s_u(k_{u2}e_u + \tau_1 + k_{u1}\dot{e_u}) \tag{4.33}$$

Hence, $\dot{V}$ is negative definite if

$$k_{u2}e_u + \tau_1 + k_{u1}\dot{e_u} \begin{cases} < 0 & \text{if } s_u > 0 \\ = 0 & \text{if } s_u = 0 \\ > 0 & \text{if } s_u < 0 \end{cases}$$

Consequently, stability can be insured if

$$\tau_1 \begin{cases} < \Gamma(e_u) & \text{if } s_u > 0 \\ = \Gamma(e_u) & \text{if } s_u = 0 \\ > \Gamma(e_u) & \text{if } s_u < 0 \end{cases}$$

where

$$\Gamma(e_u) = -k_{u2}e_u - k_{u1}\dot{e_u}$$

Hence, the required stability can be achieved through the control law

$$\tau_1 = -k_{u2}e_u - k_{u1}\dot{e_u} - k_{u3}sgn(s_u) \tag{4.34}$$

where k_{u3} >0, k_{u2} >0

The performance of the system is sensitive to the slope of sliding surface k_{u1}. A large value of k_{u1} results in a fast response but the system becomes unstable. Conversely, a small value of k_{u1} results in a stable system but the response of the system becomes slower. Thus, an optimum value of k_{u1} which results fast response and stable system is desirable. However, the existence of unmodeled system parameters makes the problem more cumbersome in determining the optimum value.

The simulink model comprising the lateral dynamics of the robot and a sliding mode roll controller of Equation (4.34) is shown in Fig. 4.9. A bias proportional

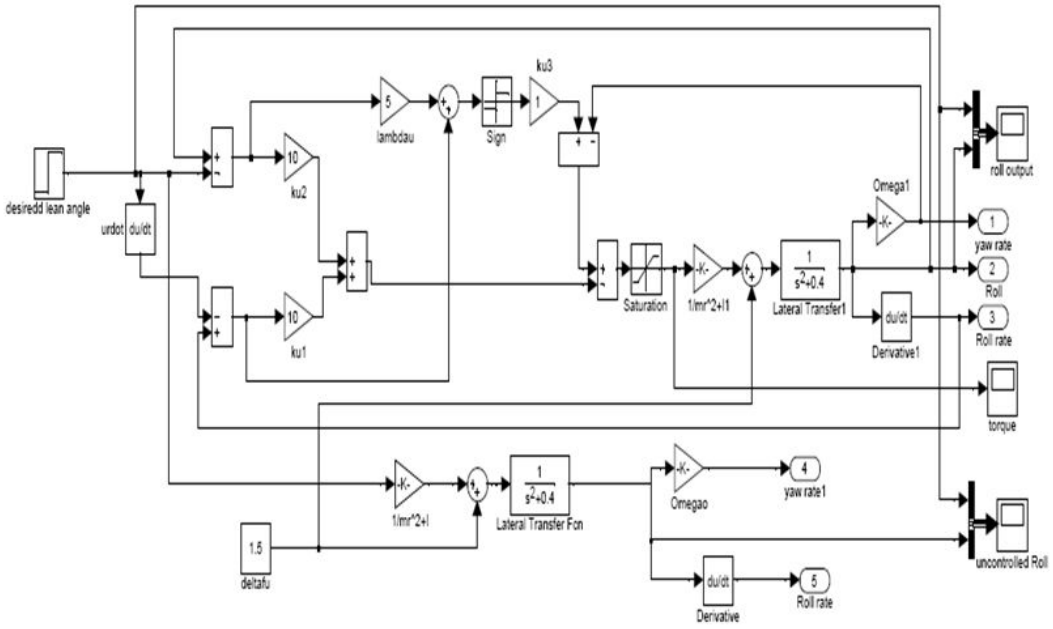

Figure 4.9: Lateral motion control simulation model

to product of nominal speed and a yaw rate of the robot is added to the sliding mode controller to compensate for the dynamic model uncertainty associated with yaw motion according to Equation (4.3). In this simulation, delay due to the feedback measurement sensors is assumed to be negligible. A chattering free and converging controller output is achieved through an experiment with $\lambda_u = k_{u1} = 10$, $k_{u2} = 10$ and $k_{u3} = 1$. The results of the simulation are discussed in Chapter 6.

This simulation is based on an approximate linear model of the prototype robot which may not represent the actual dynamics of the system. Hence, the simulation model of coupled dynamics of the robot based on Equation (3.11) is shown in Fig. 4.10. However, there are a number of factors omitted in the model to reduce complexity of the equation. For instance, friction between gears, DC motor inductance, properties of the spherical shell, etc. Hence, tuning controller coefficients experimentally with actual hardware is necessary in addition to the simulation.

4.2.3 Sliding Mode Control Implementation

The controllers designed in Equations (4.27) and (4.34) are implemented in software. Software implementation clearly requires discretization of continuous control signals. Although a sliding mode controller is designed for continuous state model, it is equally applicable for a discrete system state model with sufficiently small sampling time of the control signal. Accordingly, the continuous time derivatives of error signals are implemented as differences between samples within the loop update time. Furthermore, the set point(desired steering angle) is removed from the derivative term of lateral control Equation (4.34) since change in set point may cause an unwanted change in controller output.

One of the challenges in implementing a sliding mode controller is determining controller coefficients k_{u1}, k_{u2}, k_{u3}, k_{v1} and k_v. Determining these coefficients mainly depends on an experiment rather than simulation. Genetic algorithm is proposed to determine the controller coefficients with known upper bound (Wong and Chang, 1998). However, controller coefficients determined through simulation may not work in practice unless the system model is as accurate as the actual physical system. Hence, finding SMC coefficients which drive the states to sliding manifold with a minimum time and maintaining them without chattering is one of the most challenging experiments of this project.

However, a method which enables to determine an optimum value of slope of sliding surface k_{u1} is implemented to simplify some of these challenges. The slope of the sliding surface is continuously updated in real-time based on the system error and multiplied with predetermined value of k_{u1} to search for optimum value. The relative values of fastness and slowness of the response of the system are provided with the first and second order derivatives of the error in Equation (4.35).

$$a(k) = \begin{cases} \frac{e(k)-e(k-1)-(e(k-1)-e(k-2))}{e(k)-e(k-1)} & \text{if } |e(k)-e(k-1)| \geq |e(k-1)-e(k-2)| \\ \frac{e(k)-e(k-1)-(e(k-1)-e(k-2)}{e(k-1)-e(k-2)} & \text{if } |e(k)-e(k-1)| < |e(k-1)-e(k-2)| \end{cases} \tag{4.35}$$

The normalized acceleration of error, $a(k)$ provides a relative rate information about the system response in the range of [-1,1] where 1 and -1 represent a very fast and very slow response respectively. The medium rate that characterizes increase or decrease of system response with a constant rate takes the value of zero acceleration.

The slow and fast response of the system is controlled with a coefficient K_c which depends on error and acceleration of the error. When the error increases, K_c increases to make the system response faster and it decreases as acceleration of error increases to make the system more stable. Therefore, an empirical relation in Equation (4.36) proposed by (I. Eksin, 2002) is used to determine the value of K_c.

$$K_c = e^2 + a_{min} - \frac{(a - a_{min})^5}{1 + |e|} \tag{4.36}$$

where, a_{min} is a predetermined value for acceleration of error.

The system response is considered to be slow enough when the acceleration of error is less than a predetermined value a_{min}= 0.2. The second term in Equation (4.36) ensures that the overall slope of the sliding surface is positive. Thus, the self-tuning slope of the sliding surface of the controller with adaptive coefficient K_c is $k_{u1}K_c$.

Moreover, a saturation function in Equation (4.37) is used as a substitute of *sign(s)* to eliminate the chattering problem that exists in the practical implementation of the sliding mode controller.

$$sat(s, \delta) = \begin{cases} 1 & \text{for } s > \delta \\ \frac{s}{\delta} & \text{for } |s| \leq \delta \\ -1 & \text{for } s < \delta \end{cases} \tag{4.37}$$

where, δ is a small positive value.

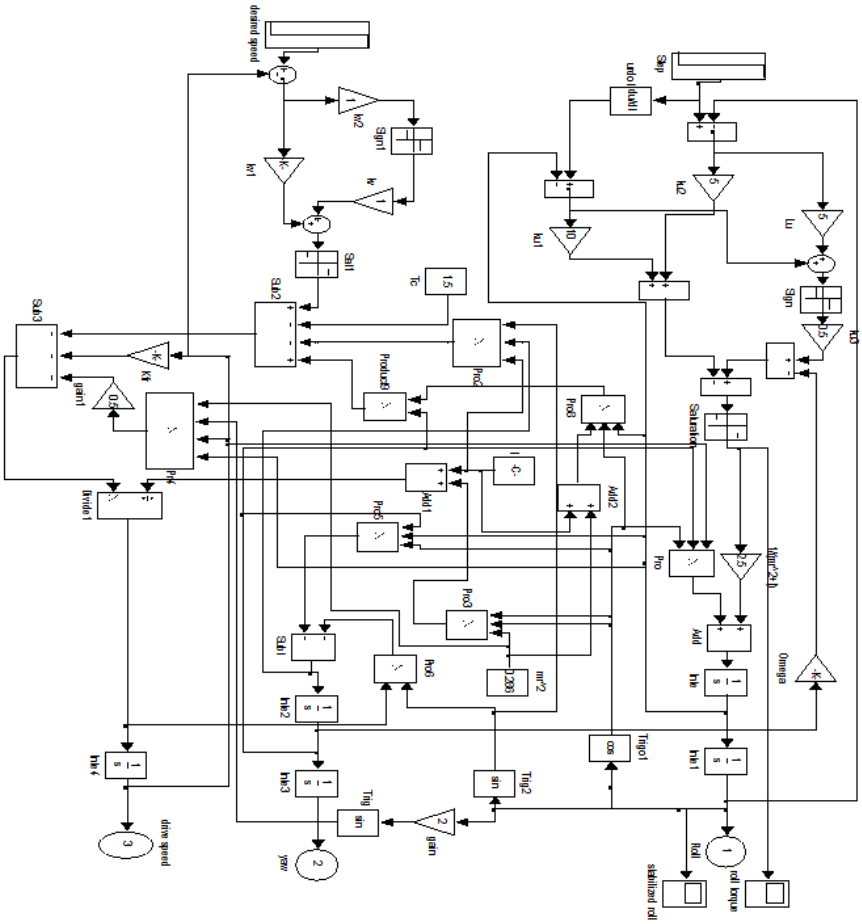

Figure 4.10: Simulation model of coupled dynamics with controller

Chapter 5

Prototype Robot Hardware and Software

The system consists of a prototype robot and control station[1]. The prototype robot is composed of a mechanical structure, electromechanical devices, electronics and software. The mechanical structure is designed and developed at the Automation and Systems Technology Department, TKK. Although the electronics schematics were designed and developed during previous research, there were several unidentified problems in the electronic circuits and servo motors. Consequently, the system was out of operation until it was resolved during this project which is briefly described in Section 5.9. More importantly, a robust control algorithm is implemented with software embedded in the microcontroller. The next subsections describe each hardware and software used for the control system.

5.1 Hardware Architecture

The prototype robot is equipped with an on-board microcontroller, a communication system, power supply, internal sensors for control and actuators as

[1] a computer used for commanding the robot

shown in Fig. 5.1. The layout of the hardware originally from (Nagai, 2008) is modified to accommodate newly implemented yaw rate sensor (gyroscope), potentiometer and servomotor. The control station performs higher level operations such as monitoring sensor data, setting speed, steering, driving and stopping the robot.

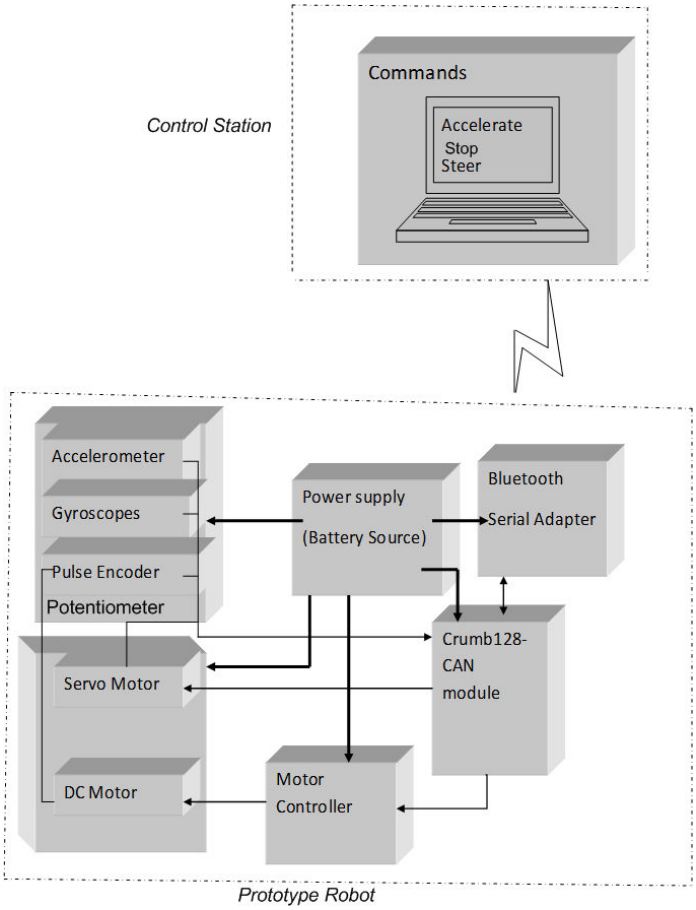

Figure 5.1: Hardware architecture

5.2 Measurement Sensors

The feedback control system requires measuring the output signals of various system parameters. Physical quantities which are measured with sensors for control are: roll angle of the robot and its roll rate, yaw rate of the robot,

tilt angle of the pendulum and its angular rate, and speed of the driving motor. The following subsections describe the sensors and their application for measurement of the physical parameters of the robot.

5.2.1 Tri-axial Accelerometer

The Kionix tri-axis accelerometer KXM52-1050 is installed to measure the tilt angle of the robot. The output response of the KXM52-1050 accelerometer depends on the direction and magnitude of accelerations. If there is no acceleration applied along an axis, the output voltage v_{off} equals half supply voltage v_{cc} with a variation of $\pm 100mv$ under a specified temperature range. When the robot accelerates in a positive direction, the output voltage increases ($v_o > v_{off}$) and it decreases ($v_o < v_{off}$) when the robot accelerates in a negative direction.

The output varies with acceleration linearly in the rate of 660mv/g according to Equation (5.1)

$$v_0 = v_{off} \pm 660mv/g \times 1g \sin u \tag{5.1}$$

where g is acceleration due to gravity. Mounting of the tri-axis accelerometer on the main shaft of the robot is shown in Fig. 5.2. The basic tilt angles can be computed from the accelerometer outputs. The computed angles are the angles that the x and y accelerometer axes make with the fixed reference xy plane. The Z-axis of a Kionix tri-axis accelerometer can be combined with the x- and/or y-axes to maintain constant sensitivity through all tilts of 360^0.

The tilt angle of the robot u according to the mounting shown in Fig. 5.3 can be obtained from Equation (5.2)

$$u = arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \tag{5.2}$$

where, a_x, a_y and a_z are gravitational accelerations in x, y and z directions respectively.

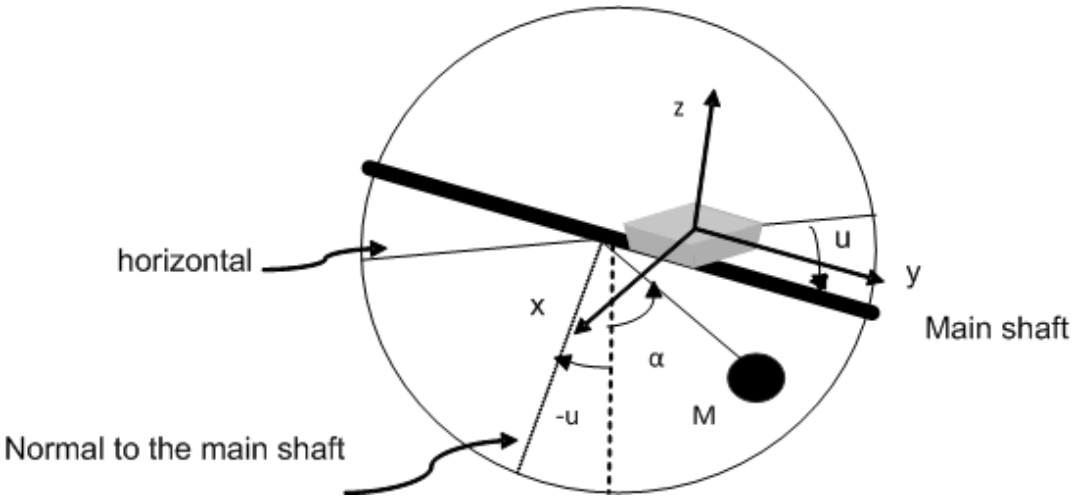

Figure 5.2: Accelerometer mounting for tilt measurement

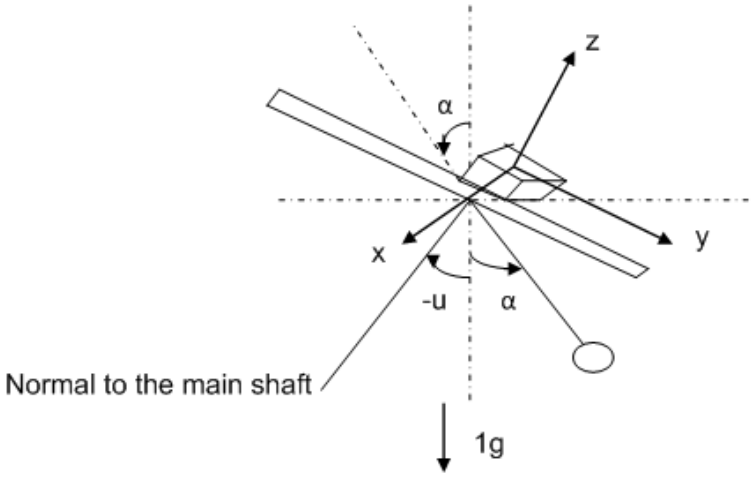

Figure 5.3: Tilt angle measurement

For a small tilt angle, the accelerometer output is proportional to the roll angle of the robot as in Equation (5.3)

$$u = arcsin(\frac{v_0 - v_{off}}{sensitivity}) \tag{5.3}$$

Hence, angular orientation is measured with accelerometer in gravity field. An experiment is performed under static acceleration (gravitational acceleration) on the prototype robot to test the performance of accelerometer for roll angle measurement shown in Fig.5.4. The figure (5.4) indicates only the average value of a particular tilt angle corresponding to output of the accelerometer. A typical tilt angle measurement for actual zero degree tilt angle after filtering is

shown in Fig. 5.5.

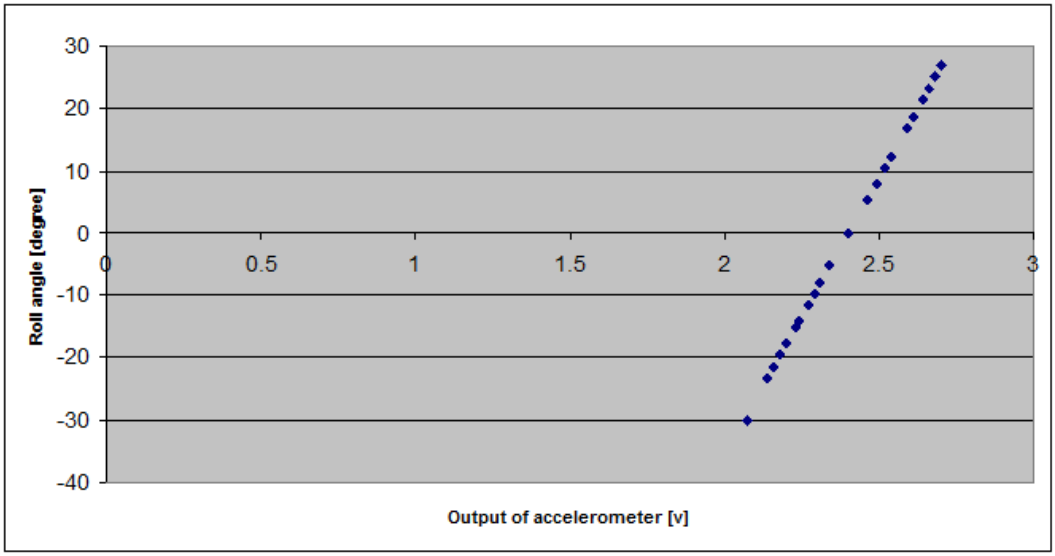

Figure 5.4: Roll angle measurement from accelerometer

In practice, the accelerometer measures both dynamic accelerations (linear accelerations of the robot and vibrations) and gravitational acceleration. Thus, the calculated inclination angle from the accelerometer is quite inaccurate particularly for this robot for which there exists no stable platform during its motion. A tilt angle measurement while the robot is accelerating is shown in Fig. 5.6. The measured tilt angle is far from the actual tilt angle of zero degree at high speed resulting from acceleration. This makes it challenging to measure the tilt angle using accelerometer. However, a better accuracy can be achieved by combining the accelerometer and angular rate measurements. Furthermore, the oscillation of the robot is relatively small at high speed due to gyroscopic stabilization. Thus, unlike high speed operation, an accurate measurement of tilt angle is necessary for low speed .

5.2.2 Gyroscopes

Gyroscopes are used to detect the angular rate of the tilt angle (roll rate) of the main shaft and angular rate about the vertical of the robot (yaw rate). They

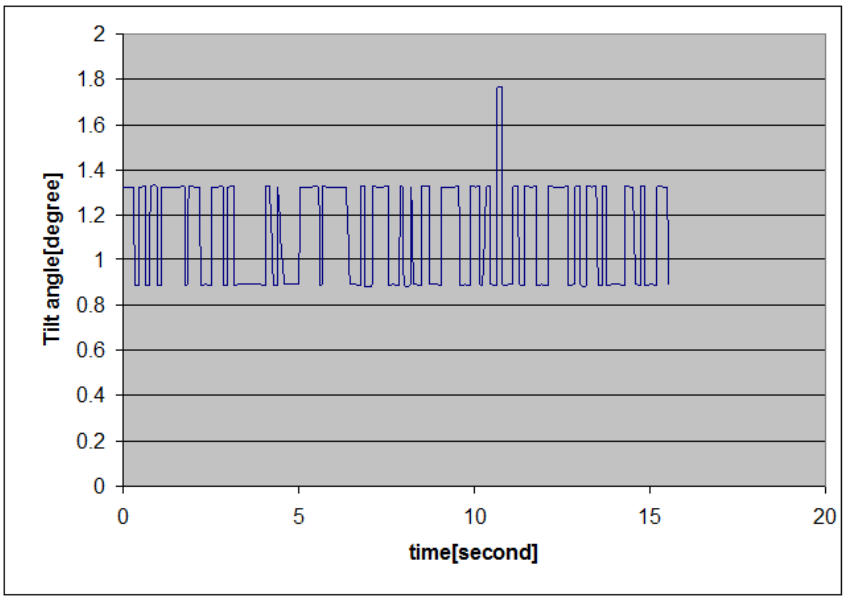

Figure 5.5: Measured tilt angle for actual zero degree tilt angle of the main shaft of the robot

measure how quickly the roll and yaw angles turn by appropriate mounting of the gyroscopes on their respective axes. The Kondo KRG-3 gyro which is a single axis angular rate measuring gyro is used for the measurement of roll rate although its sensitivity is low. Analog devices' ADSXRS300 gyro is used to measure yaw rate of the robot motion. Mounting of the two gyros are shown in Fig. 5.7. Some of the performance specifications of the Kondo KRG-3 gyro are

- Supply Voltage : 3-5V
- Output : Analogue 0.67mv/(deg/sec)
- Angular Rate Range :±300deg/sec

The output analog voltage v_0 is proportional to the roll rate of the motion of the robot along the main shaft. The sensitivity of the device is the proportionality constant which relates electrical output to the rate of lateral motion. However, the offset should be added to the relationship as in (5.4)

$$v_0 = 0.67mv \times \dot{\theta} \pm V_{off} \tag{5.4}$$

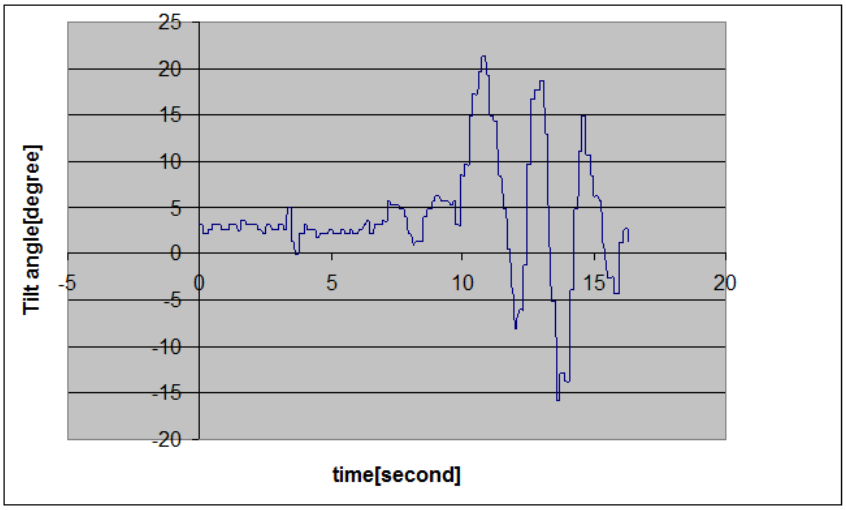

Figure 5.6: Tilt angle measurement at speed of 1.5 rps to 25 rps

where $\dot{\theta}$ is angular rate in deg/sec.

The analog voltage output of the gyros is converted to digital using a 10-bit A/D converter. The smallest value that can be recorded with the 10-bit A/D converter and 5v reference voltage is 4.88v. Hence, the roll rate gyro with sensitivity of 0.67mv/deg/sec cannot detect angular rates less than 7 deg/sec.

5.2.3 Potentiometer

The servomotor has a built-in potentiometer in its internal circuit to measure the angular position for the feedback control system. The shaft of the servo motor is coupled with the central tap of the potentiometer which moves when the servo rotates. Thus, a variable resistance and voltage proportional to the position of the servo can be obtained. Two terminals of the potentiometer are extended and fed to the A/D converter to compute the angular position of the servo with respect to the local vertical.

Moreover, the angular rate of the servo motor is calculated from the angular displacements using a microcontroller timer. Different voltages of potentiometer output are measured against the servo position for calibration as shown in

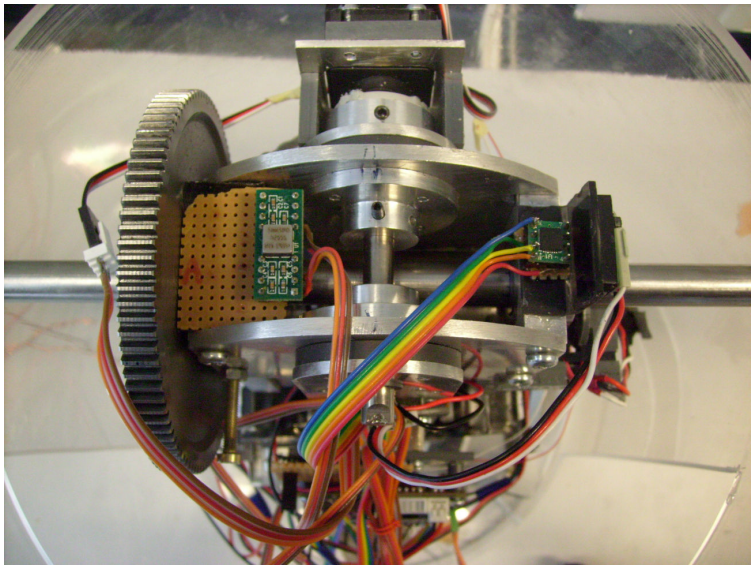

Figure 5.7: Mounting of the gyroscopes: yaw rate (Top left) and Roll Rate(Top right)

Table 5.1. The servo position 1(left) is measured as -38^0 and servo position 13(right) is $+37^0$ from the 0^0 center position 7(vertical). Thus, a random of thirteen positions in an increasing order are taken to determine the relationship between the output voltage and angular position of the servo. The output voltage of the potentiometer depends on the supply voltage of the potentiometer. During this measurement, the supply voltage of the potentiometer and the servo was 4.84v. In order to use the potentiometer as an accurate sensor, the supply voltage must be kept constant throughout its operation. For this reason, a 5v regulator and additional battery are added to the main board.

5.2.4 Pulse Encoder

A pulse encoder is used to measure the speed of the DC motor based on the number of pulses intercepted while the motor is running. The optical sensor and its peripheral circuitry provides 0v when the light is blocked and 3.5v when unblocked. The rotating disk has 36 evenly spaced black opaque marks printed on transparent plastic (Fig. 5.8) and generates thirty six pulses per rotation. The microcontroller constantly counts the pulses and increments measurement

Table 5.1: Position of the servo and potentiometer output voltage

Servo position	Potentiometer voltage [v]
position 1(Left)	1.43
position 2	1.4
position 3	1.3
position 4	1.2
position 5	1.18
position 6	1.17
position 7(Vertical)	1.15
position 8	1.1
position 9	1.01
position 10	0.95
position 11	0.84
position 12	0.8
position 13(Right)	0.79

parameter every rising and falling edge of the pulse. Thus, it produces a total of 72 ticks per rotation of the motor during the rising and falling of the pulse. The manufacturer and product name of this pulse encoder is unknown and it is a modified version of the one used in the previous projects.

5.3 Stabilized Tilt Angle Measurement

Tilt measurement from the accelerometer is a static measurement where gravity is the acceleration being measured. However, braking, accelerating and turning the robot produces acceleration. This acceleration is considered as noise since only gravitational acceleration is required to measure tilt. Thus, a tilt sensor will provide inaccurate angle measurements when subjected to motion.

An angular rate sensor may help to compensate for the effect of the tilt by measuring rotation around the proper axis of the main shaft. Unfortunately,

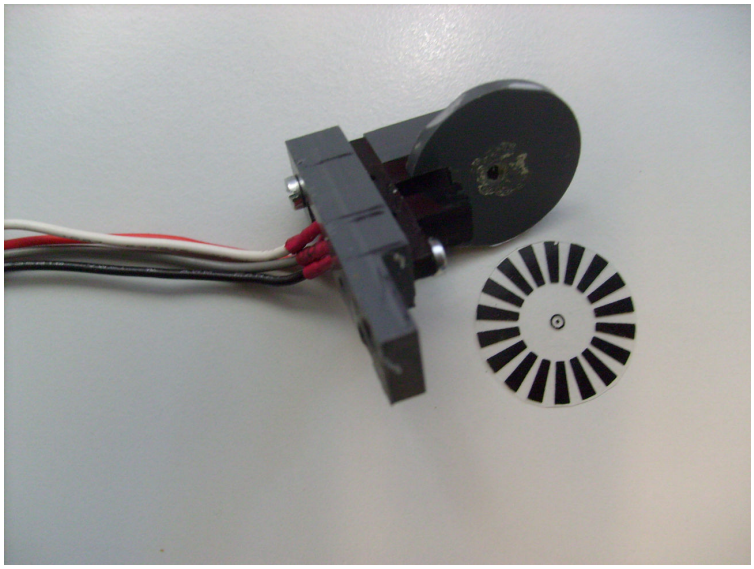

Figure 5.8: Pulse Encoder for speed measurement

the rotation angle found by integrating the measured rotation rate over time adds up to a larger error. Moreover, the offset error of the gyro which drifts over time produces an error in angle, which increases linearly with time.

However, better accuracy can be achieved by combining acceleration measurement with angular rate measurement. The angular rate sensor's accuracy over short time periods and the accelerometer's stability over long time periods can be utilized. Thus, the weaknesses of each measurement technology are compensated for by the other.

One of the techniques of sensor fusion is shown in Fig. 5.9. The difference between the two angles, *u1* and *u2* computed from a gyro and accelerometer respectively, is weighted with the gain K. Thus, the difference is the error signal which can be used to compensate for the angle calculations. The gain K controls how much of the error signal must be used to correct the angle from the gyroscope. Finally, the weighted error signal is summed with the rotation angle. The output of this process is the stabilized tilt angle predominantly from the angular rate sensor information over short time scales but corrected by the accelerometer data over long time scales.

The value of K determines the time constant at which the angle calculation

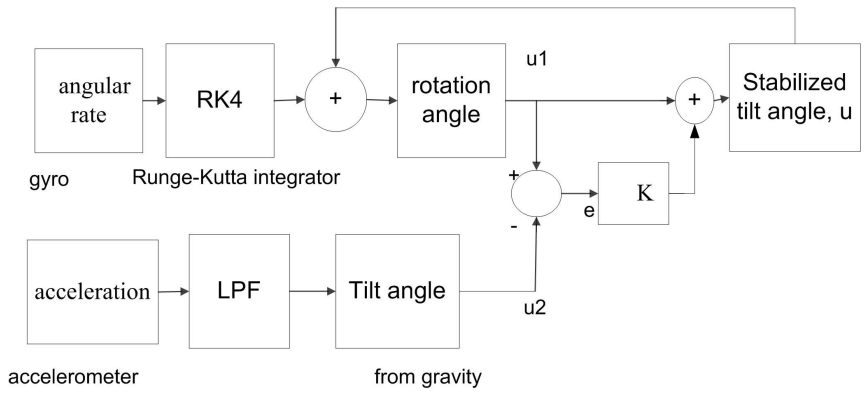

Figure 5.9: Stabilized tilt angle computed from gyro and accelerometer

from the gyro is stabilized by the gravity angle calculation. It depends on the time constant and frequency response of the sensors. According to Crossbow[2] Inertial Systems, it is recommended to take a time constant that is longer than the expected maneuvers in testing. The ratio of this time constant and measurement rate of accelerometers and gyro provides the value of K. Accordingly, the value of K = 3/50= 0.06 is chosen for the measurement rate of 50Hz.

5.4 Crumb128-CAN Module

The Crumb128-CAN module (Fig. 5.10) combines an Atmel AT90CAN128 AVR 8-bit microcontroller, a standard serial port with an RS232 transceiver and a USB2.0 device interface. It is the main on-board computer which receives commands from the control station and performs computation of the control algorithm, converts the analog sensor signal to digital and sends the corrected command to the actuators. The main features of the module used in this project are

- 128Kbytes of on-chip, non-volatile flash memory for program storage
- Dual programmable Serial Universal Synchronous-Asynchronous Receiver /Transmitter (USART)

[2]Inertial sensor manufacturing company

- 8-channel 10-bit analog-to-digital converter (ADC)

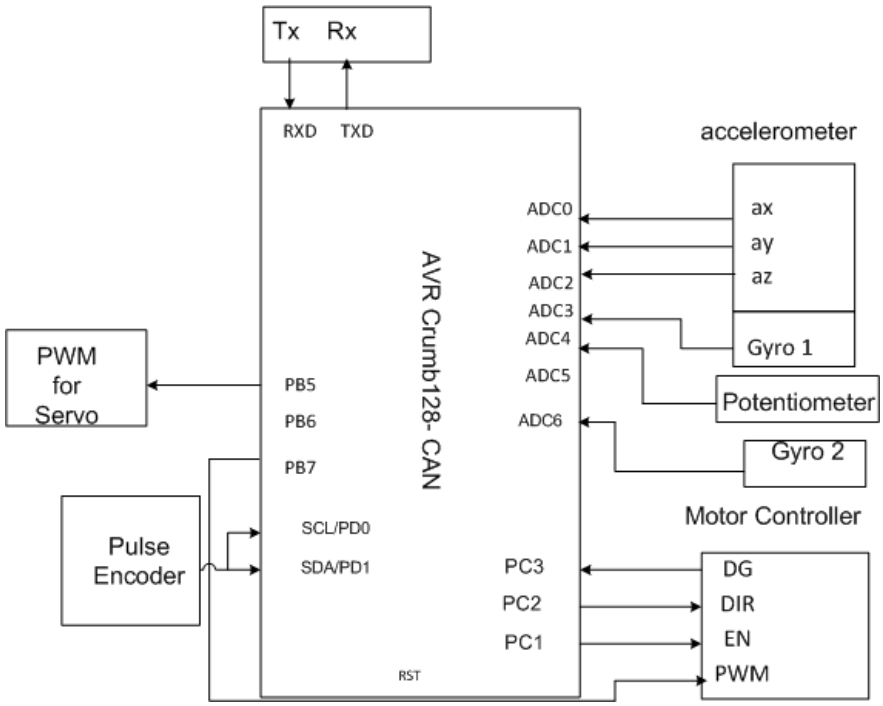

Figure 5.10: Crumb128-CAN module

Thus, a built-in 10-bit A/D converter with a 5v reference voltage enables to detect a resolution of 4.89mv from each sensor output.

5.5 Communication

An IOGEAR Bluetooth Serial Adapter GBA301 is used for wireless communication with the control station. It allows the sending and receiving data through a wireless connection of RS232 serial equipment within a range of 100 meters. The Serial Adapter is connected to the USART of the microcontroller with DB9 female connector. The data transfer rate of 9600 baud is selected and provides the best quality of transmission. The robot and control station communicate with each other through a simple communication protocol.

5.6 Actuators

5.6.1 DC Motor

The torque required to drive the robot to forward and backward is provided by a Mabuchi RS540SH DC motor (Fig. 5.11). Some of performance specifications of the motor are

- Nominal Voltage :12v
- No load speed : 17500rpm
- No load current : 0.95A

Figure 5.11: Mabuchi RS540SH DC motor

5.6.2 Servo Motor

The tilting of the pendulum is accomplished with a servo motor mounted along an axis perpendicular to a shaft axis of the DC motor at the center of the sphere. A metal gear Futaba S5301 servo motor (Fig. 5.12) which provides a torque through the arm length of the counter weight pendulum is used for the prototype robot. The main performance specifications are

- control system: +pulse width 1500us neutral
- voltage: 4.8-6v
- speed: 0.29 sec at 4.8v, 0.23 sec at 6v
- stall torque: 16.8kg-cm at 4.8v, 21kg-cm at 6v
- Operating angle: 45^0 one side pulse traveling 400us
- Direction: CCW/pulse traveling 1520-1900us

Figure 5.12: Futab S5301 servo motor

5.7 Motor Controller

A motor controller is used as an interface between the microcontroller and the DC Motor. The supply voltage to the DC motor is varied to control the speed and the resulting torque. Hence, the motor controller receives PWM and ENABLE signal from the microcontroller. A standard motor controller (Fig. 5.13) developed at the Automation and Systems Department, TKK is used. The controller inputs are

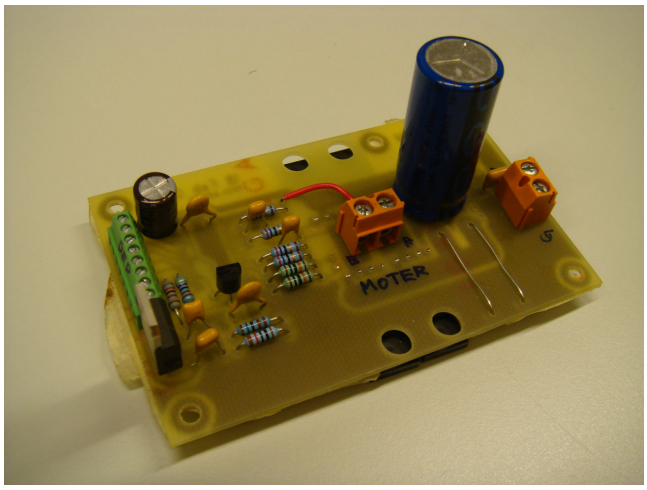

Figure 5.13: Motor controller

- PWM: Maximum frequency of 30KHZ
- Enable:To enable the control(0v or 5v)
- Direction:To set the direction of rotation (0v or 5v)

5.8 Power Supply

The robot is powered by low self-discharge NiMH batteries (LSD NiMH). The batteries have significantly reduced self-discharge, and hence are ready-to-use after charging. Furthermore, they have a reduced memory effect, allowing charging them even after slight discharging (Nagai, 2008).

Sixteen 1.2V NiMH battery cells produce a total of 19.2V. The power is distributed to different devices through a switch. The main switch is supplied with two 6 A-A battery boxes and 4.8V from one 4 A-A battery box. A 7.2V is provided to the motor controller, 4.8V to the servo motor and the remaining 7.2V is provided to the main board containing Crumb128-CAN through a 5.0V regulator.

5.9 Major Hardware Problems Encountered

At the outset of the thesis proposal, the previous hardware and main electronic circuit board were assumed to be fully operational. Therefore, only a supplementary electronics was expected to be added during this project. Thus, the main focus was to design and develop control methods to follow the desired trajectory with minimum possible oscillation which is a challenging problem to date for spherical robots. Unfortunately, the existing hardware and electronics had a number of unidentified problems. Hence, during this project hardware problem identification has been performed with a number of experimental testing procedures. The following subsections briefly describe each of these problems and their solutions associated with the servo motor and electronic circuit.

5.9.1 Servo Motor Malfunctioning

Two servo motors BlueBird BMS660MG-HS were installed to steer the ball during the previous project. These motors were badly heating when the robot was brought to operation. However, the heating reduced when mechanical fittings to the body of the shaft were dismantled and reinstalled again. Then, the ball was operated for a few minutes for demonstration purposes. Later on, the heating problem reappeared and continued before the servo motors came to an end of an operation.

One of the servo motors was uninstalled leaving only one servo for steering. An experiment was performed to determine how much one of the servos could lift the counter weight pendulum(Fig. 5.14). This enables the determination of the torque capacity of a low torque servo to operate the steering of the ball. Hence, the limit angle that the pendulum can raise the load was determined as follows.

- The maximum desired tilt angle of the main shaft of the robot from horizontal was measured

- The pendulum angular position from a vertical was measured until it did not lift counter weight anymore, which is the limit angle.

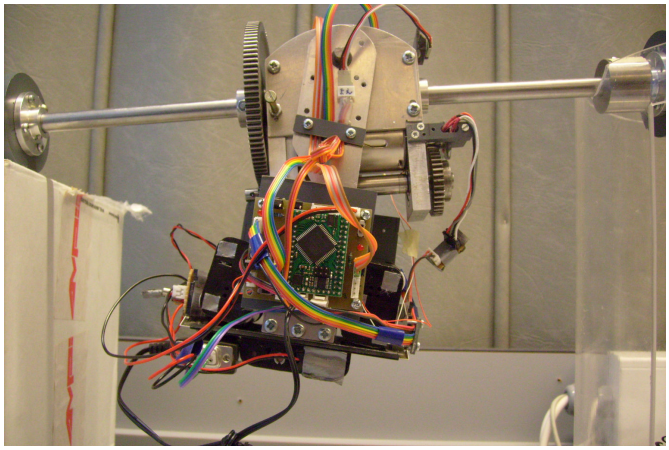

Figure 5.14: Experimental set up for determining tilting capacity

The servo motor was found to raise the pendulum to 32^0 from the vertical when the main shaft was tilted at 24^0 from the horizontal. This is quite enough to steer the robot on flat surfaces. Finally, a test for driving and steering was performed with a single servo driven pendulum. It worked well for a number of days until it stopped operation totally.

Further observation and troubleshooting of the servo was required. As a result, a broken mechanical gear was found during checking its internals which ends the use of these servos in the system. Hence, a new high torque Futaba S53o1 servo motor is installed and tested with a Radio Controller (RC) to ensure that the new servo is operational. The question why the previous servo motors failed may be raised. There may be a number of technical reasons. To mention a few, operating two servos in parallel requires synchronization of PWM pulses to drive the common load to the desired position. If this not the case, the servos misalign and oppose each other which causes the motors to draw more current resulting in heating. This is the most probable cause of the malfunctioning of the servos provided that they were not broken mechanically during mounting. Also, this misalignment may cause main electronic circuit malfunctioning due to more current drawn from the shared power supply.

5.9.2 Data Acquisition and Processing Board

The data acquisition and processing board (Fig. 5.15) is the main circuit board of the control system. It consists of a Crumb128-CAN, interface of the sensors, signal processing and control software embedded in a microcontroller. A PWM pulse is generated from this board to control the servo and DC motor which indirectly controls driving and steering motion of the robot.

When steering was performed with a new servo motor loaded with a counter weight pendulum, it oscillated uncontrollably. This might indicate that the most probable problem could be due to insufficient power to the servo motor. Hence, an external power was provided to the servo and a PWM signal was sent via a wireless link. Unfortunately, it did not work with this test. Furthermore, the ground of the external power was connected to the ground of the main circuit board to be certain that the system was in a common ground. The persistence of the problem led to check whether an appropriate PWM pulse required to the new servo was generated. Using an oscilloscope, the PWM pulse was monitored and measured as 20ms which is a standard servo pulse repetition period needed to keep the servo in the desired position. Moreover, the pulse was varying its width properly at the commanded position.

The findings of the result were puzzling and not at all what would have been expected. Hence, all the measures taken so far were summarized and plans were made for the rest of the fault finding procedures. Accordingly, it was ensured that the new servo was operational by testing with the RC. Also, the PWM signal was tested with another low torque servo with an external load rather than a counter weight pendulum and found to work properly.

Following these, an important experiment was performed to check how the PWM pulse to the servo looks like while the servo was operating with counter weight pendulum load. To do these, a PWM signal output supplied to the servomotor was monitored with an oscilloscope. Finally, an interesting result was observed in that the signal level of the PWM was much less than 1v when it was measured. Thus, the PWM signal level was vanishing when servo was

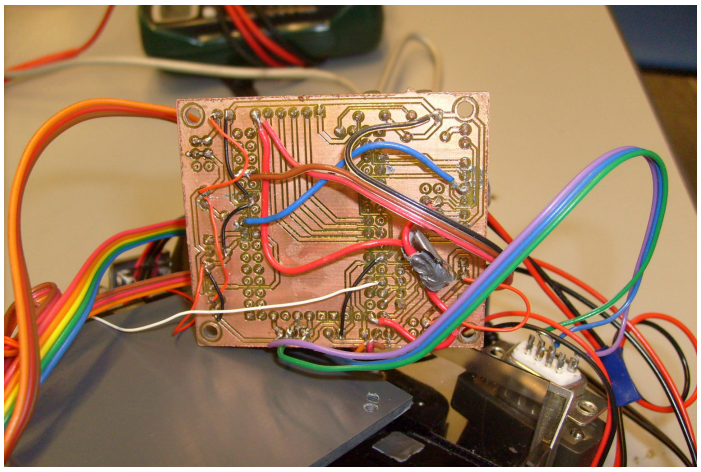

Figure 5.15: Data acquisition and processing board

driving counter weight pendulum load.

Eventually, it was concluded that there could be either a short circuit or open circuit on the main board which caused the signal level of PWM to drop when a load was given to the servo. Hence, full attention was given to the PCB of the main circuit board which consists of the wiring of interfacing sensors, bluetooth and power lines to Crumb128-CAN. Based on the schematic diagram, the circuit was checked carefully for continuity using a multimeter. And voltage levels of each power supply to the servo, accelerometer, gyroscope and Crumb128-CAN were measured to make sure that an appropriate power was being delivered. Unfortunately, these tests did not help much in solving the problem. Next, the printed circuit board was observed with a magnifying glass. However, it was not easy to identify any kind of open circuit or short circuit.

It was at this time that it was decided to write a test program which could position the servo to six different positions periodically without user control. Fortunately, it was working properly and it was a good clue to conclude that the problem was with the bluetooth communication. However, the fact that the other low torque servo which worked properly when a command signal was sent from the PC, disproved the claim of a communication problem.

In another finding, an improperly etched PCB ground line was found on the board. This ground line was easily connecting and disconnecting in an inde-

terministic manner when the pendulum was in motion. It was then replaced with an external wire and a servo motor was tested again with hope of a better performance. In spite of the PWM signal level improvement when a servo was loaded, the oscillation never stopped. However, it was one of the important achievements for the next fault findings. From these experiences, all suspected etched PCB lines were replaced with external overhead wirings. Although the servo motor stopped the continuous oscillation, it started going to the left and right end slowly without control.

At that moment, the author had no idea what to do better than those experiments to solve the rest of the problem. Nevertheless, all suspicious wirings and headers were systematically replaced in order to troubleshoot the problem. Hence, the bluetooth serial adapter wiring and other suspecting signal and power lines were desoldered and replaced with new ones to ensure that the earlier connections were correct. Consequently, a slightly better performance of the system was achieved through replacement of the wirings. Fortunately, the problem was able to be solved by a proper grounding of the circuits and rebuilding the headers of the different components. Finally, it was concluded that an electrostatic discharge (ESD) as well as improper wiring caused the system to fail. ESD is one of the most common causes of electronic failure, difficult to identify and time consuming unless preventative measures were taken during assembly and operation of the system.

5.10 Control Software

The operation of the robot is based on software. The software is implemented both in the microcontroller and the PC. Consequently, the control algorithm, sensor data acquisition and processing, wireless communication via bluetooth and PWM signal generation are implemented with an on-board microcontroller and partly with a PC for user interface. Some of the on-board software and user interface such as those handling motor controller, joystick and graphics are adapted from (Nagai, 2008) and modified.

5.10.1 On-board Software

The main tasks accomplished with the on-board AT90CAN128 microcontroller include A/D conversion of analog sensor output, generation of PWM signals to control the servo motor and DC motor, time measurement for the pulse encoder and other events, computation of the control algorithm, transmission of the sensor data to the control station and reception of commands from the control station via the bluetooth serial adapter.

Moreover, outputs of analog signals from sensors are filtered digitally. The digital equivalent of the analog RC low pass filter recommended by manufacturers of the sensors is used to filter the noise of the sensor data. This filter takes less computation time than average filter.

For the analog low pass RC filter, the outputs of the sensor and filter are related by the equation

$$v_i = R_1 C \dot{v_o} + v_o \tag{5.5}$$

The bandwidth of the filter is determined by the value of R_1C.

Equation (5.5) can be discretized as

$$v_o[n] = b_1 v_o[n-1] + a_0 v_i[n] \tag{5.6}$$

where

$$b_1 = \frac{1}{1 + \frac{T}{R_1 C}}$$

$$a_0 = \frac{1}{1 + \frac{R_1 C}{T}}$$

where, R_1 is series resistance, C is parallel capacitance and T is sampling period. Therefore, the recursive low pass filter shown in Fig.5.16 is implemented in software.

5.10.2 Control Station Software

A user interface is implemented with Visual C# for manual control of the robot and for monitoring the sensor data graphically and numerically. The user can

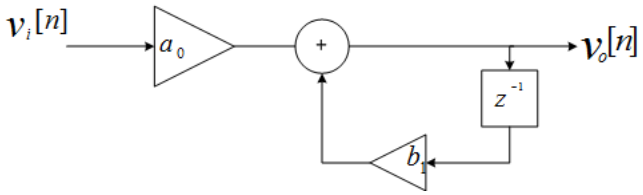

Figure 5.16: Digital low pass filter

steer and drive the robot within a region of visibility. A simple communication protocol of the format shown in Table. 5.2 is used to communicate between the robot and control station.

The control station command and on-board microcontroller sensor data transmission have their own protocols to receive each other's data. The former sends characters to signify starting of control. Similarly, a set of characters signify stopping of the robot control. Likewise, a set of characters at the start and end of a data transmission are sent from microcontroller to the control station.

Table 5.2: Data Transmission protocols

Driving and Veering command packet

's'	drive command	steer command	control coefficients	mode

Start Control command packet

's'	drive command (50)	steer command (80)	'g'	'o'		

Stop command packet

's'	drive command (50)	steer command (80)	'f'		'i'		

Sensor Data Transmission command packet

'M'	'S'	acceleration	gyro	pulse encoder	others	potentiometer	'N'	'D'

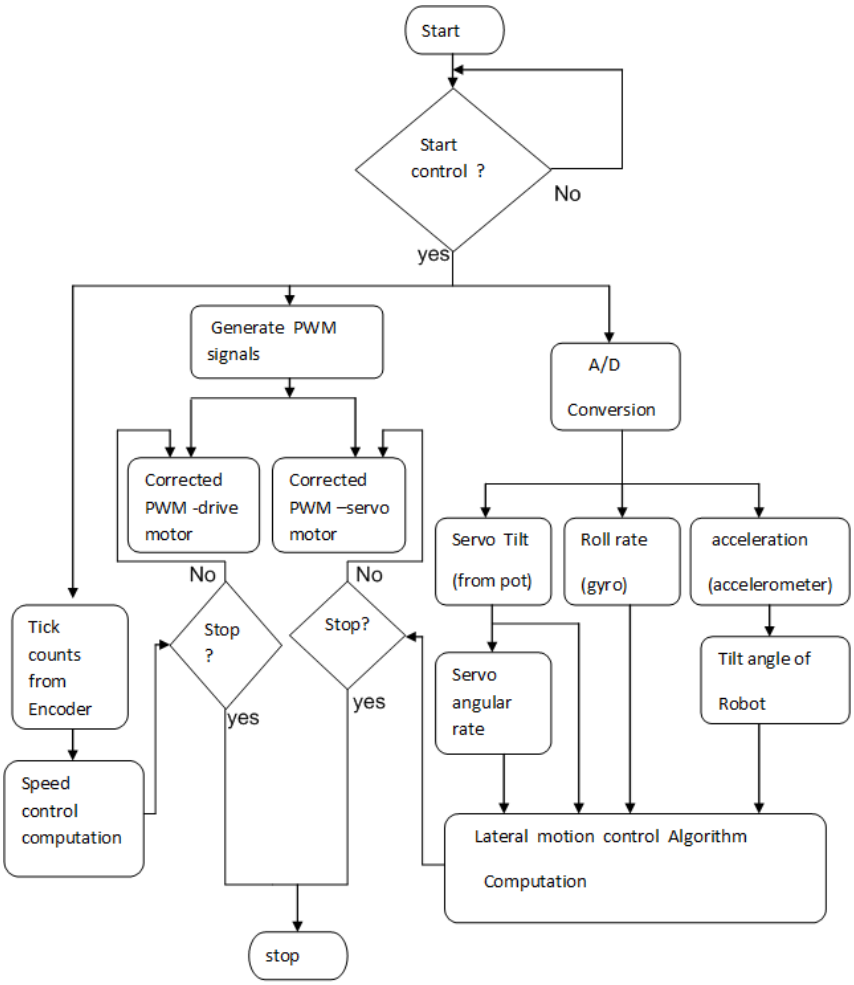

Figure 5.17: Embedded software structure excluding transmission via bluetooth

Chapter 6

Simulation and Test Results

Simulation of the dynamics of the prototype robot is performed to test the performance of the sliding mode controller. In the following subsections, the results of the simulation on the approximated linear model and nonlinear model are discussed for longitudinal and lateral motions. Moreover, an experiment is conducted on the prototype robot to verify the simulation results of sliding mode controller.

6.1 Lateral Motion

Roll angle stabilization based on linearized models in Equations (4.2) - (4.4) and Fig. 4.5 is shown in Fig. 6.3 and Fig. 6.1 with sliding mode controller and without a controller respectively. The result of the simulation in Fig. 6.1 indicates that the open loop roll motion of the robot is oscillatory with a period of about 10 seconds. Although the result is based on an approximate linearized model, the existence of the oscillation is verified with the simulation results of the actual coupled nonlinear model in Equation (3.11) and Fig. 4.10 as shown in Fig. 6.2. Moreover, similar results based on the MSC Adams model are presented in C.

Therefore, an active and/or passive damping of the oscillation is necessary particularly for autonomous operation of the robot. A simulation result of an active damping and stabilizing sliding mode controller

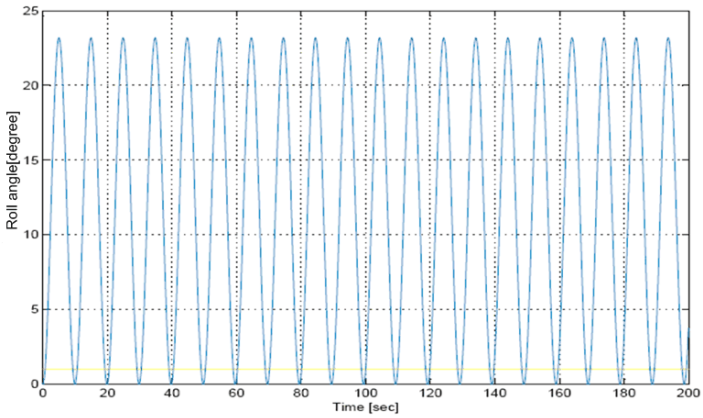

Figure 6.1: Step response of roll motion of approximated linear model without control

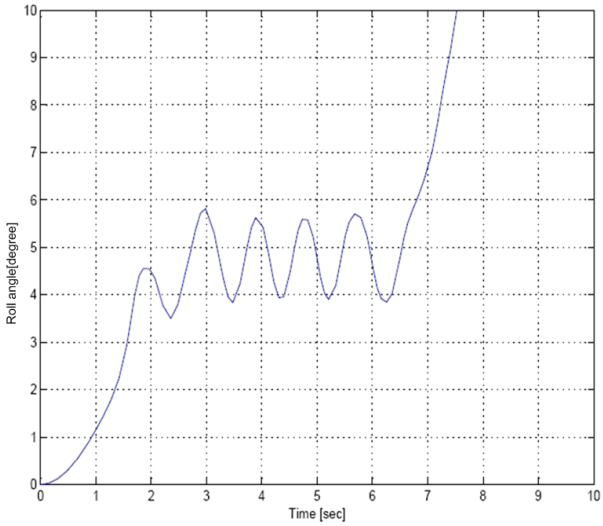

Figure 6.2: Step response of roll motion of nonlinear model without control

based on linear approximate model is shown in Fig. 6.3. The controller dampens and stabilizes the roll motion within 1 second without oscillation. Moreover, the usual problem of chattering of the sliding mode controller does not exist with control torque except for the first 1.5 seconds (Fig. 6.4). Furthermore,

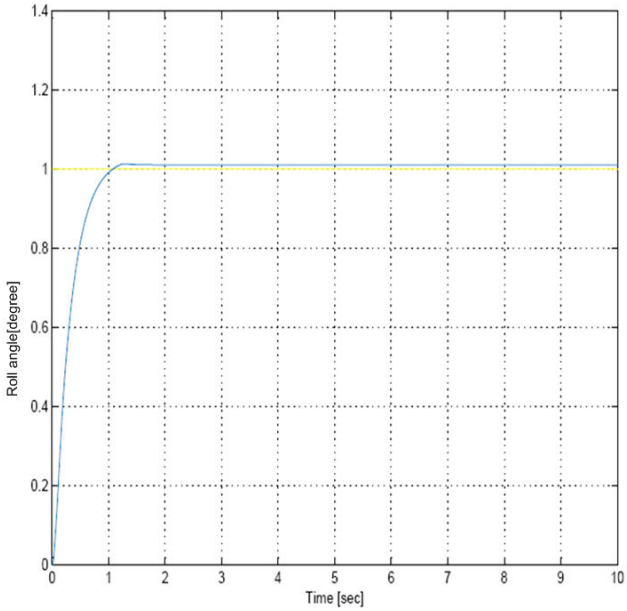

Figure 6.3: Step response of roll motion with sliding mode control

performance of the sliding mode controller is evaluated with the dynamic model corrupted by band-limited normally distributed white noise with noise power of 5 units to simulate more practical situations. Accordingly, stabilization of lateral motion is well maintained despite a slight deviation from the desired position as shown in Fig. 6.5. However, the noise model of the system and environment is not studied in this work.

The simulation result based on the approximate model is merely theoretical since linearized state equations cannot exactly represent the actual dynamics of the robot. Nevertheless, the simulation of nonlinear coupled dynamic model provides more accurate validation for the performance of the controller. Consequently, simulation of the sliding mode roll controller in Fig. 6.6 based on the nonlinear model in Equation (3.11) is provided. The controller stabilizes roll motion slowly compared to that of the approximated linear model. However, the oscillation is dampened within less than 5 seconds. Moreover, there exists no chattering with control torque output except for first few seconds. Thus, the performance of the sliding mode controller on coupled nonlinear model is close to that of the approximated model.

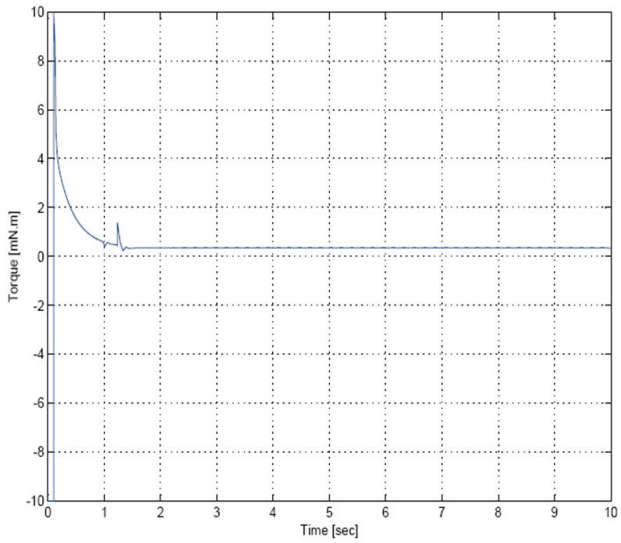

Figure 6.4: Control torque of linear model

An experiment on the prototype robot is performed to verify the simulation results. The steering angle of the robot is set to zero degrees from the vertical so as to travel along a straight line. The roll angle is stabilized after a few seconds as shown in Fig. 6.8. In another experiment, the robot is steered at constant tilt angle of 35 degree at speed of 10 RPS. This steering results in circular trajectory without oscillation as shown in Fig. 6.9. Thus, the centrifugal force and gyroscopic effect on steering provides balancing force to stabilize the motion on a curved path.

The tracking of the desired heading is shown in Fig. 6.10 which depicts roll angle and steering angle of the robot. However, it is not simple to compare tracking accuracy based on this inaccurate measurements. But, it provides sufficient information that the controller can stabilize the robot at the desired steering angle. Similar experiment is also performed to test performance of the controller supported with a passive damper on the rolling surface of the robot. The later provides better roll angle stabilization with less settling time as shown in Fig. 6.11.

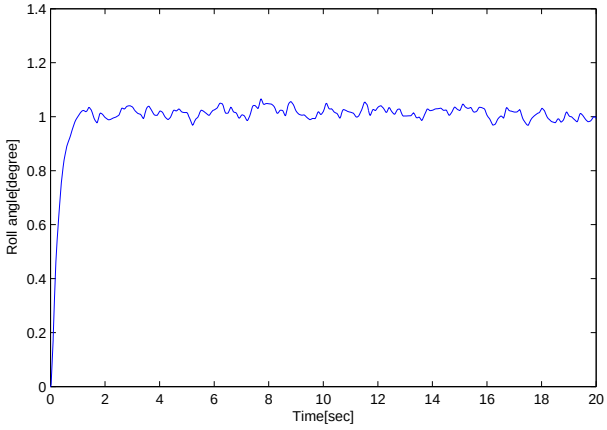

Figure 6.5: Step response of roll angle with dynamic model corrupted by white noise

6.2 Longitudinal Motion

The simulation result of longitudinal motion in Fig. 6.12 indicates that the speed of the robot can be controlled accurately without oscillation to a desired speed. Moreover, the control torque is achieved without chattering of the sliding mode controller (Fig. 6.13). This result agrees with the general fact that the pitch motion of the robot is in direct contact with the surface during rolling about the main shaft. Thus, an oscillation of the forward and backward motion is minimum. Also, the experimental result in Fig. 6.14 confirms this fact although the pulse encoder for the speed measurement is inaccurate. Moreover, speed control of the robot can easily be achieved by varying the supply voltage of the DC motor with the PWM signal. Hence, speed control is not an issue in research of spherical robots.

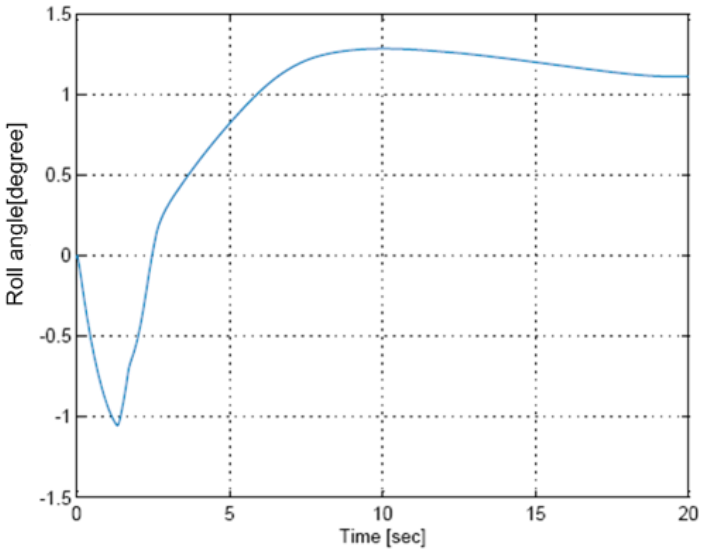

Figure 6.6: Step response of nonlinear model with sliding mode controller

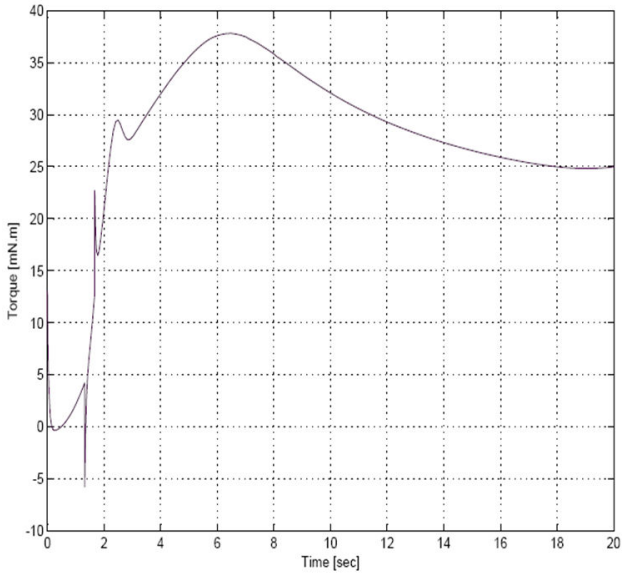

Figure 6.7: Control torque of nonlinear model

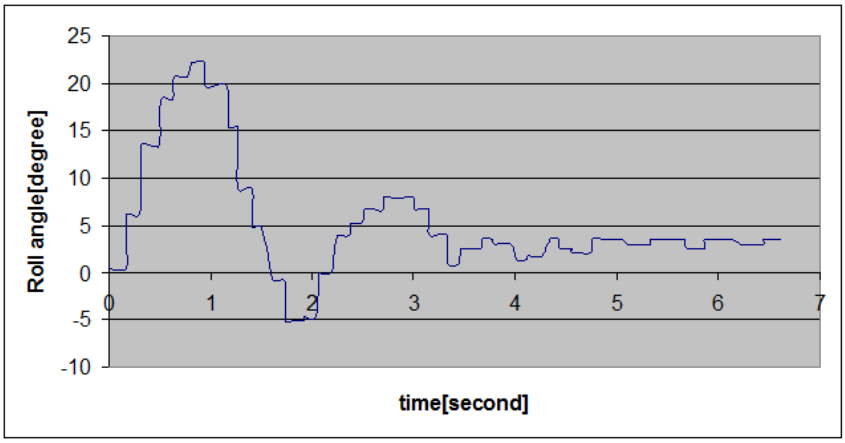

Figure 6.8: Experimental result of roll angle stabilization

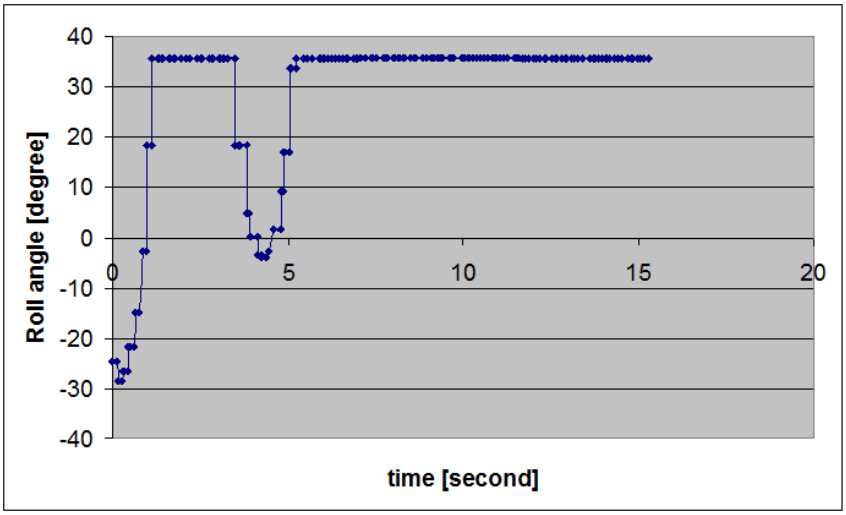

Figure 6.9: Circular trajectory at constant steering

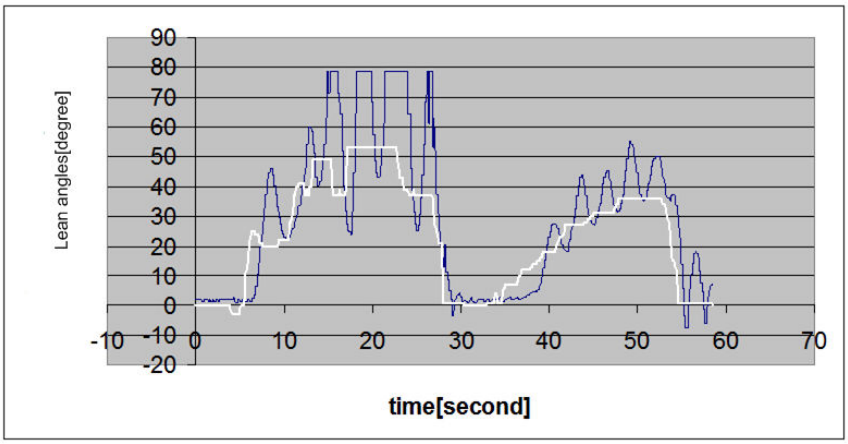

Figure 6.10: Tracking of the desired position: white-desired lean angle and black- actual lean angle

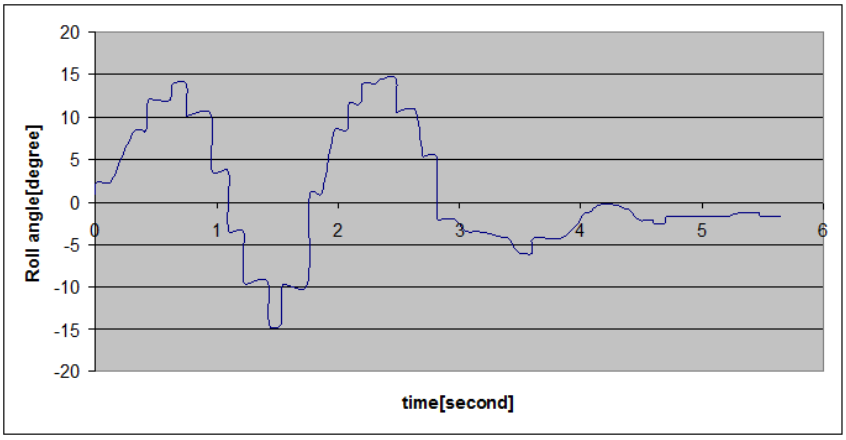

Figure 6.11: Experimental result of roll angle stabilization with support of damper

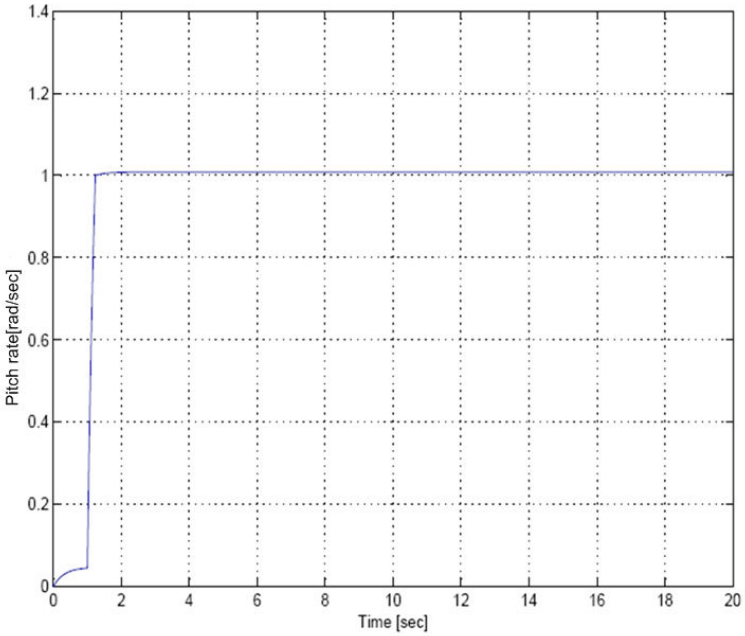

Figure 6.12: Step response of longitudinal motion with speed control

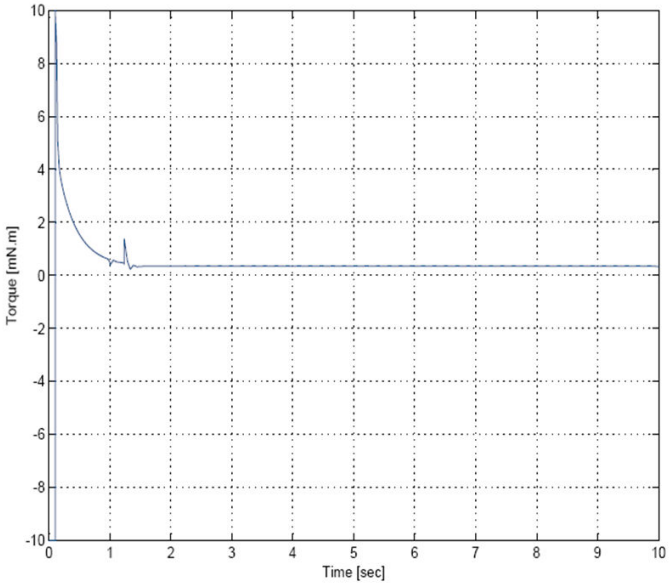

Figure 6.13: Control torque of longitudinal motion

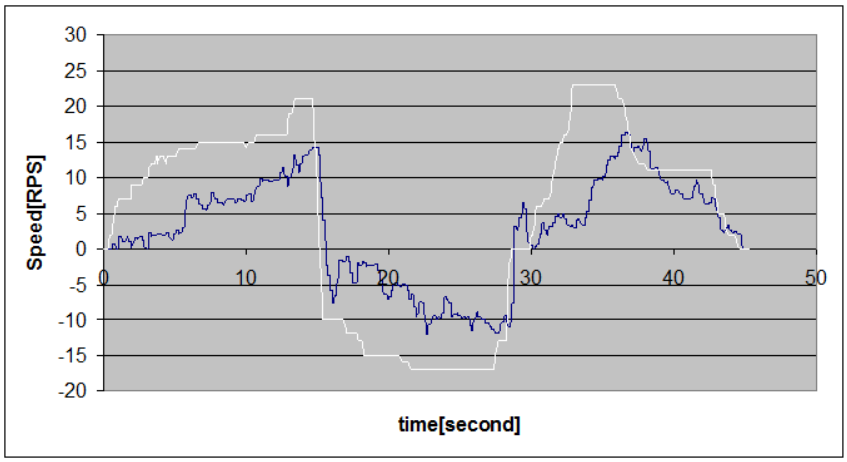

Figure 6.14: Stabilizing rotational speed of the robot; white- desired speed and black- actual speed

Chapter 7

Summary and Conclusion

The constrained kinematics which results from rolling without slipping of a spherical robot forms a nonholonomic system. Consequently, the motion of a spherical robot is highly coupled and nonlinear. Furthermore, it cannot be expressed as a chained-form system. Hence, it is impossible to directly apply both conventional control laws and nonholonomic control algorithms to eliminate lateral oscillation of the spherical robot. Thus, an effective controller design to stabilize and suppress oscillations is an essential and central point of the research into spherical robots till today.

In this thesis, dynamic model of a pendulum driven spherical robot was developed and simplified with a number of assumptions in order to control the robot to the position that is desired. The model was based on the entire dynamics of the robot in three-dimension that reveals roll, yaw and pitch motions which are crucial for designing advanced controllers. Consequently, an approximate decoupled linear model which maintains controllability of the states was used to develop robust control algorithm near operating points.

A robust sliding mode control was designed and implemented to reduce sideway oscillation of the prototype robot based on the model developed. More importantly, an adaptive sliding surface which selects an optimum slope of the sliding manifold based on the error and acceleration of the error was im-

plemented. Based on the discussion of the previous chapter, simulation and experimentation results indicated that the sliding mode controller is an effective controller for suppression of oscillation. However, the structure of the robot as well as inaccuracy of the sensors limited the performance of the controller. Hence, with further improvement of the inner mechanism of the robot, the oscillation can be avoided completely with the sliding mode controller.

7.1 Future Work

Controller design for complete suspension of oscillation of a spherical robot by itself is a research topic which requires ready-made platform with accurate sensors, reliable electronics and communication. Moreover, the mechanical structure of the robot may be modified for passive damping of the oscillation so as to reduce the control challenges. Furthermore, built-in servo control system unnecessarily complicated the control system of the robot. Hence, it is preferable to replace it with DC motor. Thus, the sliding mode controller will be directly provided to the DC motor without inner control loop. Also, the sliding mode controller may be extended to hybrid fuzzy sliding mode controller to utilize the superior benefits of both fuzzy control and sliding mode control.

References

Armour, R.H. and Vincent, J.F. (2006). *Rolling in Nature and Robotics: A Review. Journal of Bionic Engineering*, 3:195–208.

Bhattacharya, S. and Agrawal, S. (2000). *Design, experiments and motion planning of a spherical rolling robot.* In *Proc. IEEE International Conference on Robotics and Automation ICRA '00*, volume 2, pages 1207–1212. doi:10.1109/ROBOT.2000.844763.

Cameron J M, B.W.J. (1997). *Modeling mechanisms with nonholonomic joints using the Boltzmann-Hamel equations. International Journal of Robotics research*, pages 47–59.

CHEMEL B., M.E. and H., S. (1999). *Cyclops: miniature robotic reconnaissance system.* IEEE International conference.

Halme, A., Schonberg, T., and Wang, Y. (1996). *Motion control of a spherical mobile robot.* In HUT, editor, *Proc. 4th International Workshop on Advanced Motion Control AMC '96-MIE*, volume 1, pages 259–264. doi:10.1109/AMC.1996.509415.

I. Eksin, M. Guzelkaya, S.T. (2002). *Sliding Surface Slope Adjustment in Fuzzy Sliding Mode Controllers.* In *Proceedings of the 10th Mediteranean Conference on Control and Automation-MED2002.*

Javadi, A.H. and Mojabi, P. (2002). *Introducing August: a novel strategy for an omnidirectional spherical rolling robot.* In *Proc. IEEE International Conference on Robotics and Automation ICRA '02*, volume 4, pages 3527–3533 vol.4. doi:10.1109/ROBOT.2002.1014256.

Jia, Q., Sun, H., and Liu, D. (2008). *Analysis of Actuation for a Spherical Robot*. In *Proc. IEEE Conference on Robotics, Automation and Mechatronics*, pages 266–271. doi:10.1109/RAMECH.2008.4681363.

Li, Z. and Canny, J. (1990). *Motion of two rigid bodies with rolling constraint*. volume 6, pages 62–72. doi:10.1109/70.88118.

Liu, D., Sun, H., and Jia, Q. (2008a). *Stabilization and Path Following of a Spherical Robot*. In *Proc. IEEE Conference on Robotics, Automation and Mechatronics*, pages 676–682. doi:10.1109/RAMECH.2008.4681358.

Liu, D., Sun, H., Jia, Q., and Wang, L. (2008b). *Motion control of a spherical mobile robot by feedback linearization*. In *Proc. 7th World Congress on Intelligent Control and Automation WCICA 2008*, pages 965–970. doi:10.1109/WCICA.2008.4593051.

Michuad, F. and Caron, S. (2002). *Roball, Rolling Robot*. Technical report, Laborius, Department of Electrical and Computer Engineering, Universite de shebroke , Canada.

Mukherjee R, Minor M A, P.J.T. (2002). *Motion planning for a spherical mobile robot: revisiting the classical ball-plate problem. Journal of Dynamic Systems,Measurement and control*, pages 502–511.

Nagai, M. (2008). *Control system for spherical robot*. Master's thesis, Helsniki University of Technology.

Qiang Zhan, Zengbo Liu, Y.C. (2008). *A Back-stepping Based Trajectory Tracking Controller for a Non-chained Nonholonomic Spherical Robot. Chinese Journal of Aeronautics*, pages 472–480.

Rotundus (2008). *Applications of service robot Groundbot*.
URL: *http://www.rotundus.se/company2.html*

SONY (2008). *SONY Quansi-stable Traveling Action RObot*.
URL: *http://www.sonyaibo.net/aboutqtaro.htm*

Vujanovic, B. and Atanackovic, T. (2003). *An introduction to modern variational techniques in mechanics and engineering*.

Wong, C.C. and Chang, S.Y. (1998). *Parameter Selection in the Sliding Mode Control Design. Tamkang Journal of Scinece and Engineering*, 1, No.2:115–122.

Ylikorpi, T. (2005). *A Biologically inspired rolling robot for planetary surface exploration. Degree of licentiate of Technology,*. Master's thesis, Helsinki University of Technology, Department of Automation and Systems Technology, Automation Technology Laboratory.

Zhan, Q., Zhou, T., Chen, M., and Cai, S. (2006). *Dynamic Trajectory Planning of a Spherical Mobile Robot.* In *Proc. IEEE Conference on Robotics, Automation and Mechatronics*, pages 1–6. doi: 10.1109/RAMECH.2006.252705.

Appendix A

Parameters of the Prototype Robot

Parameters	Values	Symbols
Total mass	1795g	M
Moment of inertia (around rotation axis)	$0.007428 kgm^2$	J_2
Distance between ball center and pendulum CG	0.065m	l

Table A.1: Specifications of the pendulum

Determination of torque and speed constant:

$$k_t = \frac{\tau_{stall}}{i_{stall} - i_{free}}$$

$$k_e = \frac{v - i_{free}R}{\omega_{free}}$$

where, i_{free} is no load free current.

Table A.2: Specifications of the ball

Parameters	Values	Symbols
Mass of right hemisphere	985g	
Mass of left hemisphere with plastic stripe	1230g	
Left plastic (connecting main shaft and ball)	96g	
Right plastic (connecting main shaft and ball)	106g	
Total mass (excluding pendulum)	3294g	M_1
Moment of inertia (around rotation axis)	$0.0633 kgm^2$	J_1
Radius	0.226m	r
Moment of inertia of the sphere including inner mechanism	$0.11457\ kgm^2$	J
Friction coefficient(plastic shell/surface floor)	0.57	k_f
Driving gear(pinion) of first stage	10	N_1
Driven gear(intermediate shaft) of first stage	42	N_2
Driving gear of second stage	10	N_3
Driven gear of second stage	103	N_4

Table A.3: Specifications of the DC motor

Parameters	Values	Symbols
Mass	158g	
No load speed	17500rpm	ω_{free}
Stall torque	230mNm	τ_{stall}
Stall current	37A	i_{stall}
Torque constant	0.00638 Nm/A	k_t
Speed constant	0.00638 Vs/rad	k_e
Terminal resistance	0.324 Ω	R

Appendix B

Main Board Description

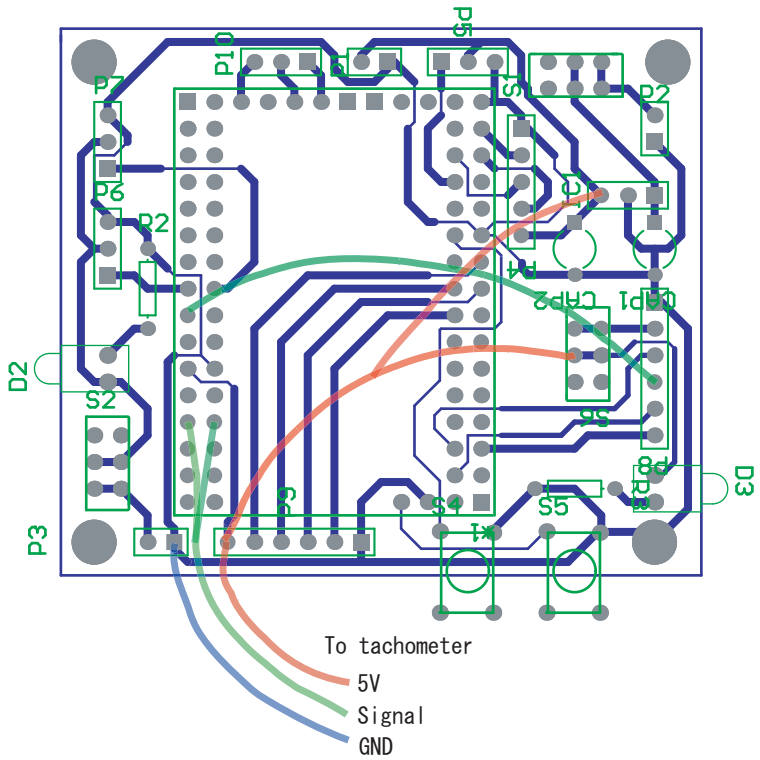

Figure B.1: Original Main Board Layout

Main board layout has some modifications of wiring and switches which can be easily identified. Moreover, the yaw rate gyro and potentiometer wiring are not shown here.

Component Name	Description	Location
AVRCrumb128	Main uchip	
4.7uF	Capacitor (electrolytic)	CAP1
0.01uF(103)	Capacitor (ceramic)	CAP2
LED	Servomotor ON/OFF indicator	D2
LED	Main board ON/OFF indicator	D3
L7805CV	5.0V Regulator	IC1
2-pin connector	5.0V for Bluetooth Adapter	P1
2-pin connector	Input Power Supply 7.2V	P2
2-pin connector	Input Power Supply 4.8V	P3
5-pin connector	For Accelerometer	P4
3-pin connector	For Gyroscope	P5
3-pin connector	For Servomotor1	P6
3-pin connector	For Servomotor2	P7
6-pin connector	For Motor Controller	P8
3-pin connector	For Bluetooth-Serial Adapter	P10
300	Resistor (metal-film)	R2
300	Resistor (metal-film)	R3
6-leg mini-switch	7.2V supply -> main board on/off	S1
6-leg mini-switch	Servomotor on/off	S2
Push-switch	not in use	S4
Push-switch	For reset	S5
6-leg mini-switch	5V from motor controller => main board on/off	S6

Table B.1: Component location on the main board

Appendix C

ADAMS Model and Co-Simulation

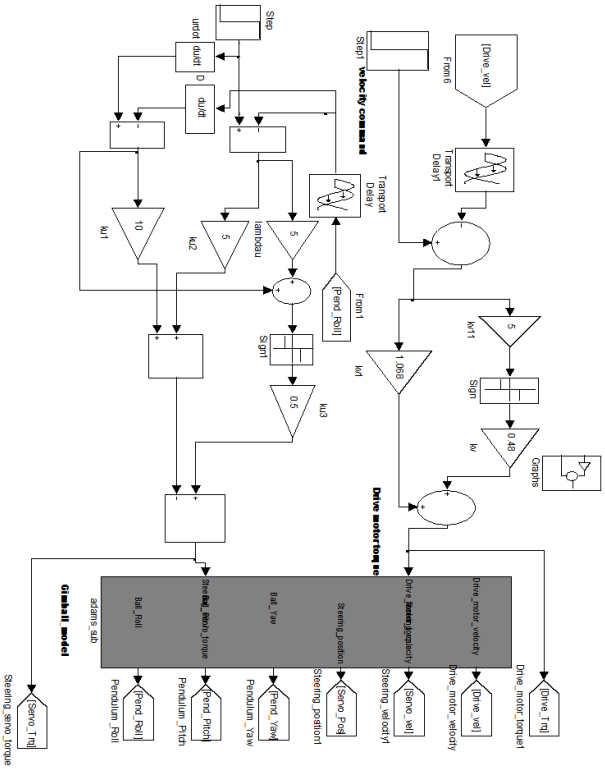

Figure C.1: ADAMS and MatLab simulink co-simulation model

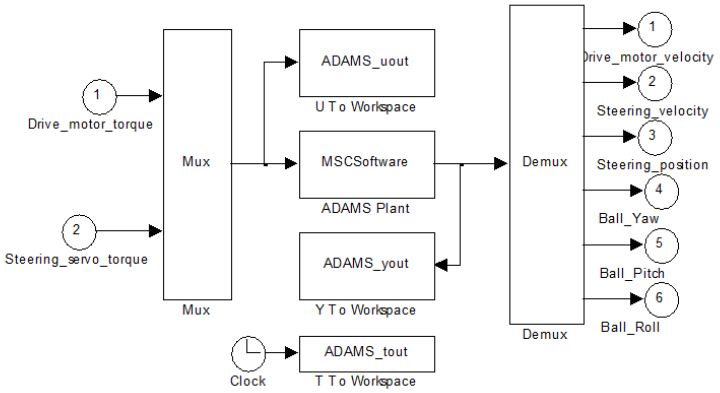

Figure C.2: ADAMS model of the mechanical system

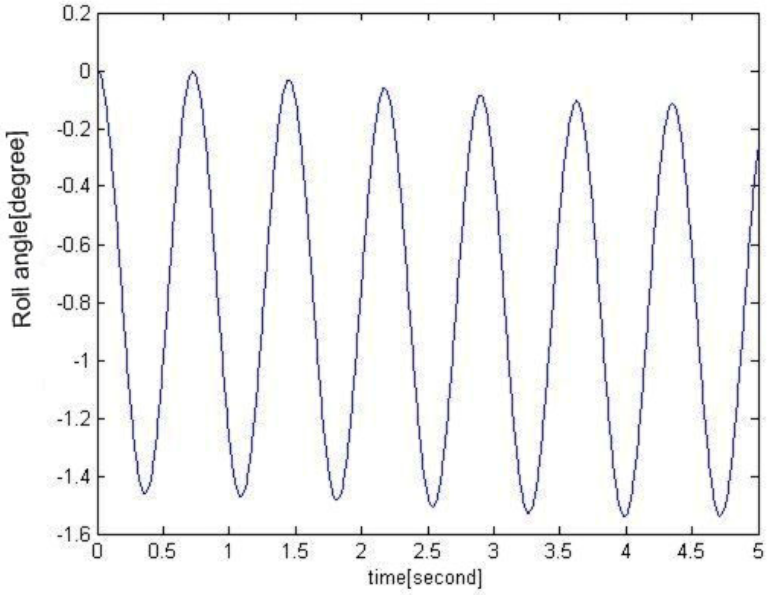

Figure C.3: Step response of lateral motion without control based on ADAMS model

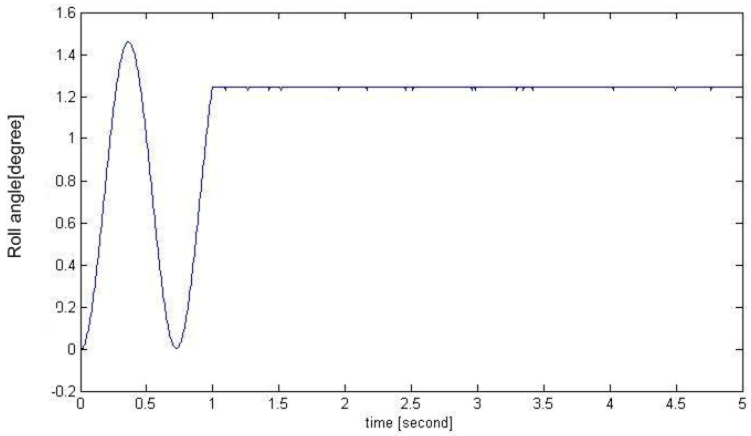

Figure C.4: Step response of roll angle with Sliding mode control based on ADAMS Model